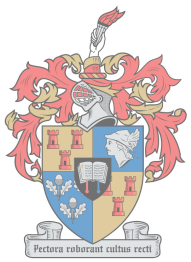
Accurate Localisation of a Multi-rotor Using Monocular Vision

by

Josua Blom



UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

Thesis presented in partial fulfilment of the requirements for the degree of
Master of Engineering
at Stellenbosch University

Supervisors:

Dr. C.E. van Daalen & Dr. P.G. Wiid
Department of Electrical and Electronic Engineering

March 2018

Plagiaatverklaring / Plagiarism *Declaration*

1 Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2 Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
I agree that plagiarism is a punishable offence because it constitutes theft.

3 Ek verstaan ook dat direkte vertalings plagiaat is.
I also understand that direct translations are plagiarism.

4 Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.

5 Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Copyrig **Studentenommer / *Student number***	Copyright **Handtekening / *Signature***
J. Blom **Voorletters en van / *Initials and surname***	01/12/2017 **Datum / *Date***

Abstract

The mid-frequency aperture array (MFAA) is planned for phase two of the square kilometre array project, which has its design phase scheduled for 2018. The MFAA's antenna arrays need to be characterised in their real world environment. The antenna array characterisation can be done with a test source mounted on a multi-rotor which is flown over the antennas. However, the test source needs to be localised accurately relative to the antenna array, which is currently achieved by expensive and cumbersome methods.

Accurate vision-based localisation is one possible inexpensive solution, provided artificial reference points can be placed in the environment. Many vision-based localisation methods exist; however, the focus is often on simultaneous localisation and mapping as opposed to localisation only. The problem is simplified significantly when artificial reference points, referred to as landmarks, are manually placed in the environment wherein the multi-rotor needs to be localised. The focus of the research presented in this thesis is therefore on accurate localisation of a multi-rotor aircraft specifically through monocular vision using manually placed artificial landmarks.

The multi-rotor's state propagation was described according to a kinematic motion model. Additionally, a measurement model was designed which relates camera image measurements to the system's states. A localisation algorithm using the unscented Kalman filter (UKF) was designed and integrated. The UKF uses the sensor data from the multi-rotor as well as measurements derived from image processing to best estimate the pose of the multi-rotor. The localisation algorithm was first tested and refined in simulation, after which experimental flight tests were performed and the resulting data sets were analysed.

The experimental results are promising; the algorithm localised the multi-rotor with a mean accuracy of around six centimetres relative to a differential GPS (DGPS) that was used as a baseline. A high quality DGPS can localise at an accuracy of up to two centimetres; however, the Piksi DGPS used in this project proved to be intermittently accurate and unreliable. The current accuracy of the localisation algorithm would be suitable for other radio telescope antenna arrays which operate at lower frequencies than the MFAA. However, with some improvements in hardware integration, it should be possible to achieve better accuracy than differential GPS systems at a fraction of the cost, making it a promising solution for localisation in antenna characterisation application on the MFAA.

Abstrak

Die mid-frekwensie stralingsvlak reeks (MFAA) is beplan vir fase twee van die vierkante kilometer reeks (SKA) projek, waarvan die eerste ontwerpsfase beplan is vir 2018. Die MFAA se antennas moet gekaraktariseer word in hul geïnstalleerde omgewing. Die karaktarisering kan gedoen word deur middel van ‘n toets-bron wat op ‘n multirotor geplaas is, en dan oor die antenna gevlieg word. Die toetsbron moet egter akkuraat gelokaliseer word relatief tot die antenna, wat tans deur duur en omslagtige metodes gedoen word.

Akkurate visie-gebasseerde lokalisering is een bekostigbare moontlike oplossing, gegewe dat kunsmatige verwysingspunte in die omgewing geplaas kan word. Hoewel baie visie-gebasseerde lokaliseringsmetodes bestaan, is die fokus meestal op gelyktydige lokalisering en kartering, eerder as slegs lokalisering. Die probleem word heelwat vereenvoudig wanneer kunsmatige verwysingspunte, wat landmerke genoem word, met die hand in die omgewing geplaas kan word. Die fokus van die navorsing wat in hierdie tesis aangebied word is dus die akkurate lokalisering van ‘n multirotor-tuig, deur die gebruik van ‘n enkelkamera en kunsmatige landmerke.

Die multirotor se toestandveranderlikes is beskryf deur middel van ‘n kinematiese beweginsmodel. Daarbenewens is ‘n meetmodel ontwerp wat die beeldmetings se verband met die toestande van die multirotor beskryf. ‘n Lokaliseringsalgoritme wat gebruik maak van ‘n ongegeurde Kalman-filter (”unscented Kalman filter” of UKF) is ontwerp en geïntegreer. Die UKF maak gebruik van sensor data vanaf die multirotor, asook van metings wat afgelei word deur beeldverwerking om die toestande van die multirotor te bepaal. Die lokaliseringsalgoritme is aanvanklik getoets en verfyn in simulasie, en daarna is eksperimentele toetsvlugte uitgevoer en die resulterende data ontleed.

Die eksperimentele resultate is belowend; die algoritme het die multirotor gelokaliseer met ‘n gemiddelde akkuuraatheid van rondom ses sentimeter relatief tot ‘n differensile GPS (DGPS) wat as ‘n verwysing gebruik is. ‘n Hoë kwaliteit DGPS kan lokaliseer tot en met ‘n akkuuraatheid van twee sentimeter; maar die Piksi DGPS wat in die projek gebruik is, het met wisselvallige betroubaarheid opgetree. Die huidige akkuuraatheid van die lokaliseringsalgoritme sal geskik wees vir ander radio frekwensie teleskoop antennas wat teen ‘n laer frekwensie as die MFAA werk. Met sekere verbeterings in hardeware integrasie behoort dit egter moontlik te wees om beter akkuuraatheid as ‘n DGPS te behaal vir baie goedkoper, wat die oplossing baie belowend maak vir toepassing in antenna karaktarisering van die MFAA antennas.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations and Acronyms

AUT	antenna under test
DGPS	differential global position system
DNSS	global navigation satellite system
EKF	extended Kalman filter
EM	electromagnetic
FC	flight controller
FMA	flying machine arena
GPS	global positioning system
IF	information filter
IMU	inertial measurement unit
KF	Kalman filter
LiDAR	light detection and ranging
MAV	micro aerial vehicle
MFAA	mid-frequency aperture array
PnP	perspective-from-n-points
PF	particle filter
RF	radio frequency
SIFT	scale invariant feature transform
SKA	square kilometre array
SLAM	simultaneous localisation and mapping
SURF	speeded-up robust features
UAV	unmanned aerial vehicle
UKF	unscented Kalman filter

Notation

x	Scalar
X	Axis
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$\boldsymbol{\mathcal{X}}$	Set of sigma points
$\hat{x}$	Estimate of x
$\bar{x}$	Average of x
σ_x	Standard deviation of x
σ_x^2	Variance of x
$\boldsymbol{\mu_x}$	Mean of $\mathbf{x}$
$\boldsymbol{\Sigma_x}$	Covariance of $\mathbf{x}$

Coordinate Systems

X, Y, Z	Inertial landmark axes
x_c, y_c	Image plane axes
X_C, Y_C, Z_C	Camera axes
x, y, z	Body-fixed axes
N, E, D	Inertal north-east-down axes

Acknowledgements

"Coming back to where you started is not the same as never leaving." - Terry Pratchett

As one my greatest achievements, this thesis would never have been realised were it not for the support and influence of so many notable people around me.

Firstly, and most importantly, none of my achievements would have been possible without the continuous and unwavering support of my parents. Despite my lack of stellar performance during my years being home-schooled and then going to school, my mother and father taught me that I can achieve anything through perseverance and hard work. I was born into extremely fortunate circumstances and a loving family, for which I am forever grateful, as there are many deserving people who live through struggles that are no fault of their own.

I would like to express my deepest and sincerest gratitude towards my supervisors, Dr Corné van Daalen and Dr Gideon Wiid. I have been blessed with their exceptional support throughout this project, and their shared knowledge was invaluable every step of the way. Thank you for your patience and your guidance.

I have to acknowledge everyone at the Electronic Systems Laboratory who has played a part in the academic adventure of my postgraduate studies. Your camaraderie and the good experiences we have shared will always be fondly remembered.

Lastly, I would like to thank my amazing wife, who has supported me through all of the frustrating and difficult times, and celebrated with me when they were overcome. I look forward to sharing many more challenges with you, and will always appreciate your loving support.

Chapter 1

Introduction

The use of multi-rotors in various industries have grown exponentially over the past few years owing to their advantages over fixed-wing and helicopter aircraft. Fixed-wing aircraft require a lot of space to operate and helicopter aircraft are expensive to operate and maintain. However, multi-rotor aircraft consist of few moving parts, do not require a lot of space to take-off and land, and are inexpensive to operate and maintain. Small remotely controlled helicopters offer the same functionality as multi-rotor aircraft, but lack benefits such as high stability, heavy payload carrying capabilities and highly advanced automated control systems. New and innovative applications come to light every year where the use of unmanned aerial vehicles (UAVs) such as light multi-rotor aircraft save costs and manpower. Most of these applications include a specialised payload-carrying multi-rotor that performs certain tasks with the payload. Alternatives to UAVs are large helicopters or fixed-wing aircraft that can be used to carry the specialised payload, but with the advancement of technology the payloads have become much lighter and smaller, which renders full-size aircraft unnecessarily expensive. The combination of advancements in UAV technology and payload designs make the application of multi-rotors a serious competitor in the aviation and UAV industry.

UAV payloads often consist of camera systems that are used to perform land surveys or take video footage. However, there are several cases where a UAV is used to carry a specialised piece of equipment for engineering and scientific applications. In one such case the specialised payload is a dipole antenna and radio frequency transmitter used in the characterisation of much larger outdoor radio antennas [1]. Mounting the payload to a UAV is a simple task, as is flying the UAV in proximity to the subject of the test since many UAV flight controllers have semi-autonomous functions by default. These flight controllers also come equipped with several on-board sensors and an estimator which provides information regarding the states of the UAV. In many applications of a UAV, it's important to have detailed information on its position and orientation, which is provided by on-board sensors to a certain degree of accuracy. However, when more accurate positional information is required it is necessary to consider other approaches. In the case of antenna characterisation, very accurate position and orientation estimates are required to obtain near-field measurements, which drives the demand for accurate localisation of a multi-rotor. The focus of this thesis is to solve the problem of accurate localisation of a payload-carrying UAV for the application of near-field antenna characterisation whilst utilising inexpensive hardware.

This chapter provides a motivation for the work performed in this thesis. Initially, a brief overview of radio antenna characterisation is given, with a introduction to how UAVs form a part of the solution. The focus then shifts towards the problem statement and the rationale behind the

need for alternative localisation techniques. A high-level proposed solution is introduced before the literature study, followed by an overview of remaining chapters.

1.1 Overview of Radio Antenna Characterisation

The mid frequency aperture array (MFAA), depicted in Figure 1.1, is planned to be phase two of the square kilometre array (SKA). The MFAA will be composed of thousands of antenna elements which are grouped in clusters, where each cluster is permanently installed inside a protective enclosure. Many of these clusters will be placed over an area of several square kilometres in a randomised pattern, owing to the nature of radio frequency interactions between the clusters. The new antennas designed for the MFAA have a simulated elemental power pattern, this pattern requires verification in order to characterise the antenna in its real-world environment. Normally, an antenna's power pattern is measured in an anechoic chamber; however, even a small sub-array of the MFAA's antennas is too large to be measured in such a way. Various outdoor near- and far-field measurement techniques exist, such as balloon- and helicopter-borne methods, but they are very expensive and cumbersome to implement. One of the solutions that has been investigated by Virone et al. [1] is to use a small multi-rotor equipped with an artificially polarised test source as payload to characterise the antenna under test (AUT) at its installed location. The multi-rotor flight path is recorded with an optical retro-reflector system and a ground station for tracking. The data collected by the ground station measurement system can be used in conjunction with the multi-rotor's flight path to reconstruct the antenna's power pattern and compare it to the simulation.

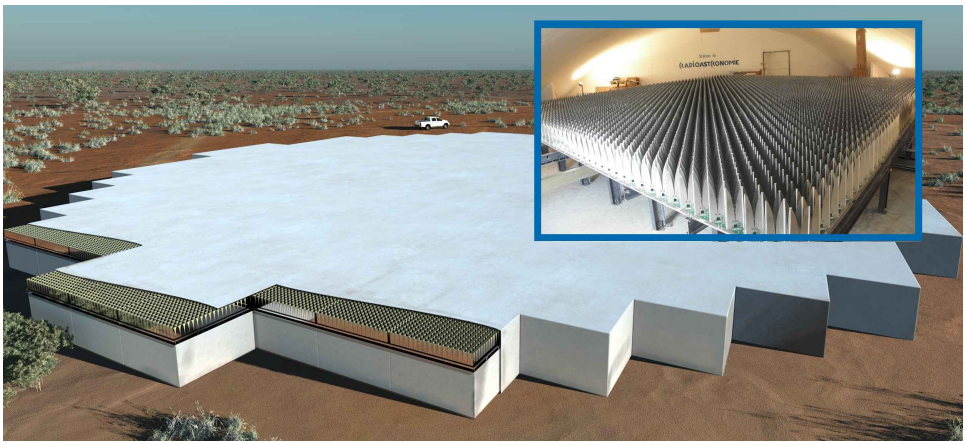


Figure 1.1: An artist's impression of a single cluster (or array) of the MFAA which consists of a protective enclosure and underlying antenna elements [2]. (Reproduced with permission)

The MFAA operates at relatively low radio frequencies of 400 to 1450 megahertz, as stated by [3]. The frequency range at which the MFAA will operate means the wavelength of the radio waves will be relatively long, ranging between 0.2 and 0.7 meters. The practical implications of wavelength on the characterisation process is the accuracy at which the test source needs to be localised. An antenna which operates at a lower frequency will have more lenient localisation accuracy requirements for the test source, whereas at higher frequencies the accuracy requirements are higher with a smaller margin for error. For frequencies below 100 megahertz a normal high-end GPS system will be sufficient for

localising the test source, considering that the measurements are taken at a higher altitude above the AUT [3]. However, when antennas with operational frequencies higher than 100 megahertz are being measured, the test source has to be closer to the antenna (under an altitude of three meters), and requires localisation accuracy of at least one centimetre [3]. The problem that is demonstrated above clearly shows the need for an accurate multi-rotor localisation method that can assist in near field antenna characterisation.

1.2 Problem Definition: Multi-rotor Localisation

The equipment used in the characterisation of the AUT consists of an artificially polarised test source, which is mounted on the multi-rotor, that transmits radio frequencies with a standard dipole antenna [4]. As the test source moves over the antenna and transmits radio waves (as depicted in Figure 1.2), its radiation pattern is picked up by the AUT on the ground, and thereby the power pattern of that antenna can be characterised. The pattern of the radio waves received by the AUT vary depending on the position and orientation of the multi-rotor. Moving the test source along a specific, predetermined path over the AUT is usually a difficult and cumbersome task. Utilising the versatility of a multi-rotor aircraft to move the test source over the AUT has simplified the characterisation process significantly. However, since the characterisation of the radiation pattern is dependant on the pose (position and orientation) of the test source, it is necessary to know the exact pose of the multi-rotor in the time domain. The problem can be solved by either precisely controlling the flight path of the multi-rotor (positioning), or by flying the multi-rotor along an approximate flight path and accurately locating it in a 3D space (localisation).

Modern off-the-shelf flight controllers for multi-rotors come equipped with several sensors which can deliver a plethora of real-time information. It is possible to interact with several states of the multi-rotor to a certain degree of accuracy, including acceleration, velocity, global position, orientation, altitude and heading. Some of these states are measured at a high degree of accuracy, which is the case for orientation. Other states such as position and altitude will have to be determined by a more accurate means, due to the inherent inaccuracy of consumer GPS systems. Off-the-shelf GPS systems used in most applications are only accurate to about three meters. To achieve higher

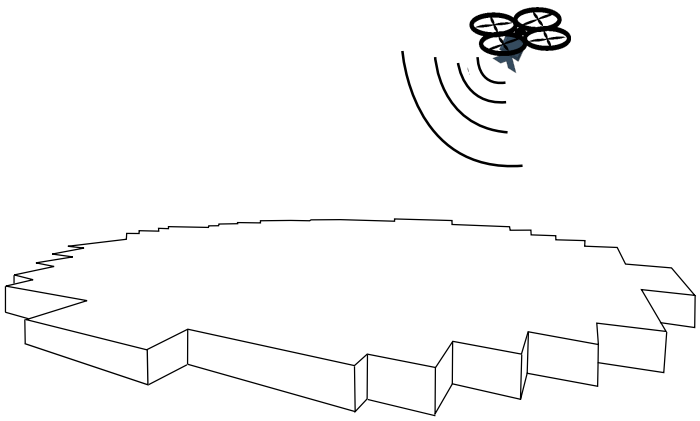


Figure 1.2: A single cluster that would form part of the MFAA is characterised by an artificially polarised test source, which is mounted on a multi-rotor.

GPS accuracy, specialised and expensive differential GPS (DGPS) equipment is required.

Accurate positional control of any UAV is a challenging task, especially when external factors such as wind and weather are present. Even though multi-rotors are substantially more agile and manoeuvrable than most other UAV platforms, no inexpensive, off-the-shelf control system is capable of controlling a multi-rotor accurately enough to meet the requirements for this problem. In this particular case the required positional accuracy will have to be smaller than one centimetre in order to characterise a 400 megahertz antenna array [3]. Taking into consideration the lack of accuracy in positional control, the alternative is measuring the position of the multi-rotor (localisation). The required localisation accuracy is achievable with DGPS and real-time kinematic (RTK) systems which allow for one centimetre accuracy at best when implemented correctly. Another alternative is an optical retro-reflector localisation system, such as the one used by Virone et. al. [1]; however, these off-the-shelf solutions are extremely expensive. The objective in this thesis project is to achieve accurate localisation whilst simultaneously minimising costs to justify an alternative to current and traditional methods. There are several other methods which can be used to localise an object in a 3D space, some of which include LiDAR, sonar, optical retro-reflectors, stereo vision and monocular vision, each of which will be briefly discussed. The method that looks the most promising with regards to low cost hardware and potentially high accuracy is considered as a candidate solution.

Light detection and ranging (LiDAR) is a surveying method that uses pulses of laser light to measure the distance to a target by detecting the reflected pulses with a sensor [5]. Many LiDAR equipment manufacturers claim that up to centimetre vertical relative accuracy is achievable [6]. In order to localise the multi-rotor relative to the AUT using LiDAR, a prerequisite would be a known map to which the LiDAR measurements can be compared. There are models of the antenna structures which can be used; however, their surfaces are large and flat, with few features that can be used for relative localisation. Features such as landmarks can be placed on the antenna, which can easily be detected by a LiDAR system. Despite the feasibility of LiDAR enabled localisation, these type of sensors are very expensive and complex to implement, especially when compared to vision-based solutions.

Sonar is similar to LiDAR, but instead of using light this method uses sound. Sonar therefore presents similar challenges to LiDAR, with the added drawback of being less accurate in aerial applications [7].

Optical retro-reflector systems are capable of localising an object with up to centimetre accuracy using a ground station that tracks a light reflector. This type of system is simple to implement; however, similar to the DGPS, it is extremely expensive [4].

Stereo vision is an image processing technique that utilises depth perception to determine the relative location of two cameras used to take a picture of the same subject. Stereo vision localisation can achieve up to millimetre accuracy; however, this varies substantially depending on the quality of the vision sensors and the distance to surrounding objects [8]. This technique requires two cameras to be mounted on to the multi-rotor, which presents challenges in itself, as well as a known map of the environment in which to localise itself. One can by-pass the requirement of a known map by manually placing landmarks in known locations in the environment. By using pictures that were taken of the landmarks, it is possible to localise the cameras relative to the landmarks with good accuracy [9].

Monocular vision can achieve the same results as stereo vision whilst using only one camera, assuming that three or more landmarks are visible in each picture. When estimating pose from a single landmark, monocular vision is not as accurate as stereo vision, however it can still achieve

millimetre accuracy [10]. While stereo vision uses two cameras, monocular vision uses a single camera and can therefore not utilise depth perception; however, when three or more landmarks with known positions are visible, it is possible to solve for the relative position of the camera. Stereo vision benefits from the ability to use any object close to the cameras as a landmark, through feature detection in image processing. However, since landmarks can be manually placed at known locations in this problem, the use of a second camera in stereo vision will not benefit the state estimation in any way, other than providing an additional image at each time step. Therefore, by using several landmarks and a single camera, the complexity of stereo vision is avoided, at a much lower monetary cost whilst also minimising computational costs.

Considering the various options, it is clear that the most suitable candidate solution is a single, inexpensive camera mounted on-board the multi-rotor which takes pictures of landmarks that are manually placed in the environment at known locations. A low cost solution is chosen with the goal of finding out how accurately it can localise the multi-rotor, and if it will be sufficient for application in antenna array characterisation.

1.3 Proposed Solution

By utilising man-made landmarks, which are strategically placed on the AUT over which the multi-rotor will fly (as shown in Fig. 1.3), it is possible to determine the position of the multi-rotor using monocular vision. The relative position can be calculated by taking pictures of the landmarks with a single, downward-facing camera and performing image processing on the pictures. The landmarks should therefore be designed to be easily identifiable by image processing techniques and each landmark should be unique. By knowing the real-world location of each landmark that is visible in the image, it is possible to determine the relative position of the multi-rotor with good accuracy [11]. This is a localisation technique that is commonly used in autonomous systems and in particular simultaneous localisation and mapping (SLAM) applications [12]. The SLAM problem is solved by using sensor and odometry data to estimate the vehicle states, whilst simultaneously estimating the landmark states in the environment to build a map of its surroundings. However, in SLAM applications, the landmarks that are used do not need to be placed at specific locations, as long as they are visible they can be used to generate measurements. The exact placement of

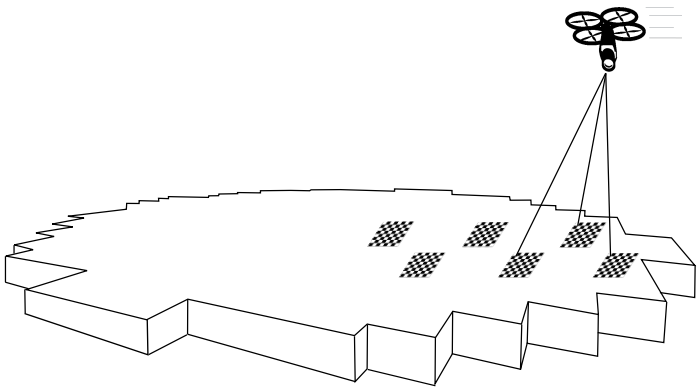


Figure 1.3: As the multi-rotor flies over the AUT, it takes pictures of the landmarks which can be used to localise the multi-rotor in post processing.

landmarks eliminates the need for implementation of a full SLAM algorithm, which can quickly become extremely complex and computationally expensive. The downside of localisation as opposed to SLAM, is that the manually placed landmarks need to be positioned very accurately.

The position tracking of the multi-rotor will be performed off-line and post-flight, since the antenna measurements will also be processed off-line. Therefore, a data acquisition system, which includes a camera and an on-board computer, will be used to capture images and other sensor data during the flight. The flight controller that is used to control the multi-rotor can also provide motion model data such as velocity and altitude. By making use of the motion model data obtained from the flight controller and the measurement model data from the image processing algorithm, a localisation algorithm can be designed to reconstruct the flight path of the multi-rotor. Figure 1.4 shows a brief overview of the necessary steps to implement this solution.

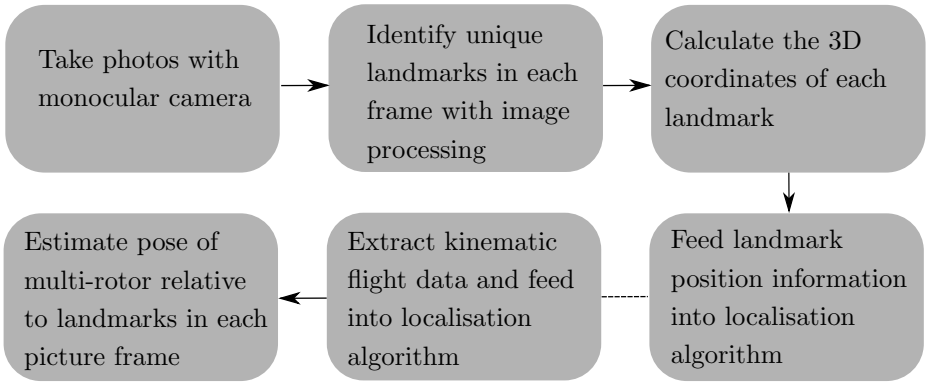


Figure 1.4: An overview of the flow of information that would take place at each time step during implementation of the proposed solution.

The photos that are taken by the monocular camera undergo image processing operations which identifies all of the landmarks inside the frame. The 3D landmark positions are then calculated by using the landmark position coordinates in the image frame as well as unique camera parameters. The localisation algorithm uses the landmark position information in its measurement model, as well as the kinematic flight data in its motion model, to determine the pose of the multi-rotor for each time step. This process is repeated for each image that was taken by the camera as the multi-rotor flew over the AUT. The result of which is a reconstructed flight path detailing the pose of the multi-rotor at each time step. All of the components that are present in the data acquisition system are shown in Figure 1.5.

1.4 Overview of Thesis

Following this opening chapter is a literature review, which details all of the underlying theory that is required to understand the system modelling, image processing and state estimation relevant to this thesis. A brief review of previous vision-based localisation techniques provides useful insight which will assist the design of a solution.

The hardware and data acquisition system is then introduced in Chapter 3, which includes a hexacopter multi-rotor platform, camera sensor and a differential GPS that is used for ground truth measurements. The modelling of the system follows in the next chapter, which initially defines

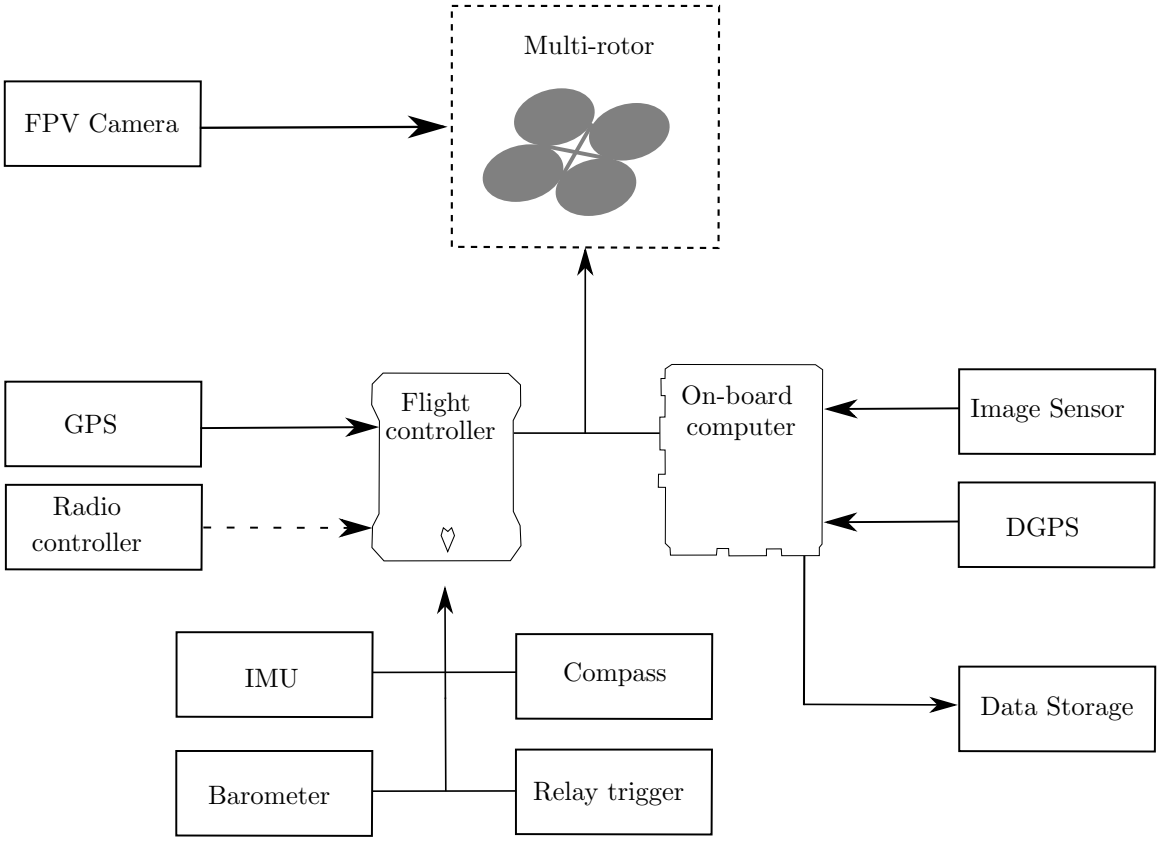


Figure 1.5: Overview of data acquisition system.

all of the state variables. The theory behind Euler axes transformations is explained, followed by a definition of important axes transformations that are performed on sensor data acquired by the multi-rotor system. The remainder of the chapter details the motion model and accompanying process noise, as well as the measurement model and measurement noise. Chapter 5 gives an overview of image processing techniques and discusses the relevant methods that are used in this thesis, including the motivation and reasoning for the checkerboard landmark design.

The next two chapters of this thesis details the design and implementation of a state estimator that is core to the localisation algorithm. The theory behind Gaussian state estimation filters is explained followed by an elaboration on the unscented Kalman filter (UKF) that is used in this thesis. The UKF algorithm and theory is discussed in a stepwise manner, after which the application specific implementation is detailed. The localisation performance of the UKF algorithm is then tested in simulation, where the filter is tuned and the results are analysed. In Chapter 8 an analyses of the practical flight test results is performed and compared to differential GPS measurements which are taken as ground truth. The localisation algorithm's performance is discussed and evaluated in its applicability to antenna array characterisation. The concluding chapter summarises the findings and experimental results of this thesis, and makes suggestions for improvements that may benefit future work on this project.

Chapter 2

Literature Review

In this literature review, the methods that have previously been used to solve the problem of accurate localisation through monocular vision is reviewed. The techniques used in those solutions are evaluated in their applicability to this problem, and the relevant topics are discussed in more depth.

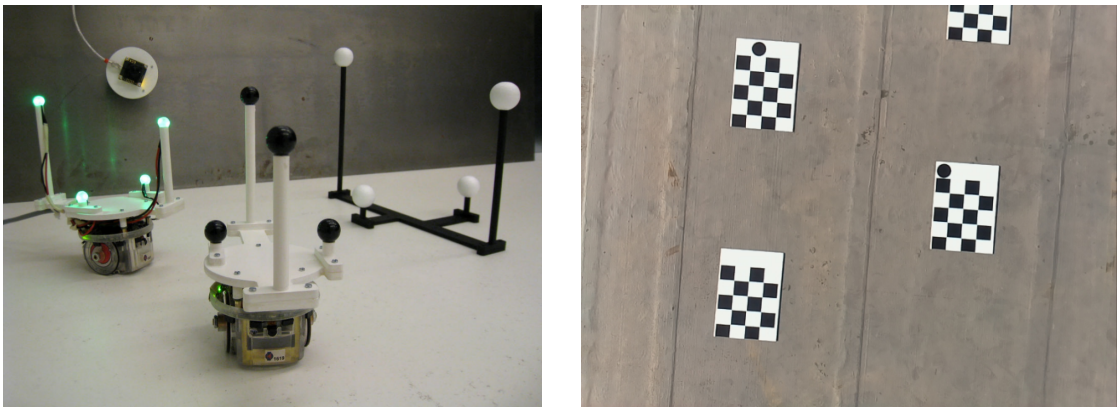
2.1 Vision-Based Localisation

One of the most popular methods of localisation in autonomous systems is simultaneous localisation and mapping (SLAM). A UAV employing a SLAM algorithm is capable of both estimating its surroundings (mapping) as well as its location within the estimated space (localisation) [13]. SLAM algorithms can be realised in several different ways by utilising various methods, some of which include occupancy grid-, fast-, DP-, EKF- and UKF-SLAM [14, 15, 16]. Each of the SLAM methods use different estimation techniques and sensors, which affect the functionality of the algorithm in a certain application. However, all of the methods make use of odometry data, as well as measurements from external sensors. The external sensors provide measurements in various forms, which may include monocular vision sensors that provide images, or LiDAR sensors which are used to build occupancy grid maps. In most SLAM applications, the measurements are used to identify landmarks, which are static points of reference that can be recognised repeatedly between several measurements. A landmark could be a point that forms part of an object in an image measurement that is taken by a camera, or it could be a recognisable feature in a occupancy grid map constructed from a LiDAR sensor [17].

In many cases implementing a functional SLAM algorithm is a challenging task, especially when there is no prior knowledge of the environment. However, the same principles and methods used in SLAM can be applied in localisation problems, particularly when certain point landmark positions in the environment are known. SLAM functionality necessitates high quality data and large quantities thereof in order to perform accurately. The algorithm has to estimate its surroundings and build a map based on that knowledge, as well as its location within that map. When new sensor information is available, it updates the estimates of its current location as well as the estimates of map. However, localisation algorithms require only information about landmark positions, assuming that there is a predefined map of landmark locations. It is clear that there is a distinct difference between SLAM and only localisation, with a substantial increase in complexity between the implementation of the two methods. The data acquisition process, however, is similar when implementing SLAM or localisation only, depending on the type of sensors that are used as well as the format of the

map. Therefore, there is a lot that can be learned from previous work done on both SLAM and localisation problems that use monocular vision-based methods.

A notable and very applicable example of previous work done on relative localisation of robotic systems using monocular vision, is that of Breitenmoser et al. [18]. Although their project focuses on the relative six dimensional localisation of three small robots, they also place a fixed landmark in the environment (see Figure 2.1a). The fixed landmark, as well as the point features on the robots, consist of small balls of various sizes that are placed in a fixed pattern. Each ball can be uniquely identified by image processing from its size, and the orientation of the landmark and robots can be determined from the position of each ball within the fixed pattern. Each robot is localised relative to one another, and relative to the fixed landmark, from an off-board externally mounted camera. In one of their final tests, a multi-rotor is used to test the relative localisation capabilities of their monocular vision and estimator algorithms. The resulting mean position error was 1.5 centimetres, measured from the ground truth, which was taken inside the Flying Machine Arena [19]. The methods they used and the results achieved are very similar to the end-goal of this project.



(a) Small robotic landmarks [18] (b) Chessboard landmarks

Figure 2.1: (a): Three robots that are localised relative to one another using monocular vision, the static landmark is visible as the black structure with white balls attached to it [18]. (b): An example of chessboard-type landmarks which are often used in monocular vision-based localisation problems [20].

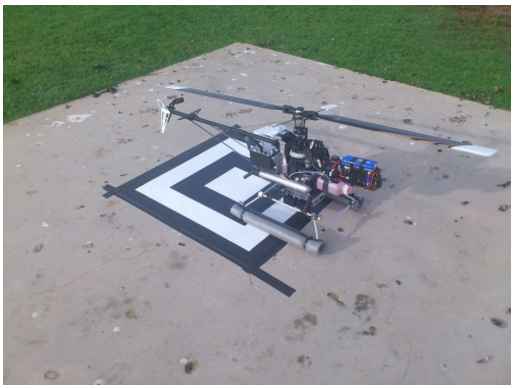
The Flying Machine Arena (FMA) is an indoor space equipped with precision motion capture equipment, and it is dedicated to the study of autonomous systems and aerial robotics [19]. The FMA is an ideal environment in which to record the ground truth of any localisation algorithm owing to its very accurate Vicon motion capture system [21]. The motion capture system consists of eight off-board, externally mounted cameras which observe ball-shaped retro-reflectors (similar to 2.1a), which are mounted on the test vehicles that need to be localised. The system can provide position and orientation data for all the appropriately marked vehicles, with up to millimetre accuracy. The Vicon system is the ideal solution for any localisation problem, however the set-up of the motion capture cameras make its application very situational, and often unsuitable for outdoor implementation.

In a paper by Altug et al. [22], an off-board vision-based control system is used on a multi-rotor for control purposes. Although this thesis focusses only on the localisation of a multi-rotor and

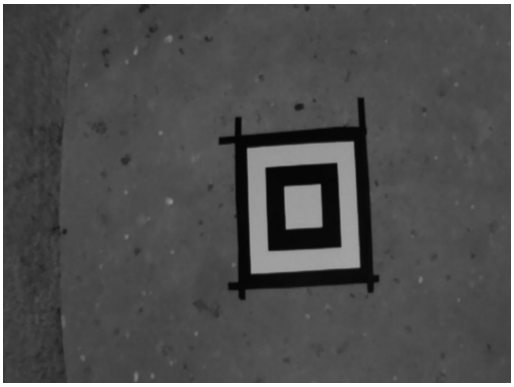
not the control thereof, the implementation of localisation techniques are required for the control of the multi-rotor relative the camera. A stationary camera is placed underneath the multi-rotor, which provides visual feedback of the states to the control system. Only the altitude and yaw of the multi-rotor is estimated, with a standard deviation of 4.2 cm in altitude, and 11.7 degrees for yaw.

In a more applicable paper by Eberli et al. [23], a monocular vision-based system is used on a micro aerial vehicle (MAV) in order to determine its pose relative to a single landmark. The vision sensor was mounted on the MAV in a downward facing orientation to capture image measurements of the landmark directly below it. The artificial landmark, which consisted of two concentric circles, was used to determine the pose of the MAV. The pose estimation algorithm could then determine the position of the MAV by comparing the current image of the landmark to a known image of the landmark. The algorithm could also estimate the orientation of the MAV whilst being robust to illumination changes and motion blur. The algorithm could estimate the position of the robot with a mean error of around eight centimetres at a height of 90 cm above the landmark, whilst no results are provided for orientation estimation.

In a novel approach to autonomous aerial navigation [13], an MAV was localised in a completely unknown environment through monocular vision SLAM. The vision sensor was mounted on-board the MAV and measurements were taken online in a live exercise, where map points were constructed from image processing operations. Each map point can be considered a landmark, which can be used for relative localisation of the MAV. The focus of that project was on autonomous and explorative navigation in an unknown environment, however the landmark detection and relative localisation techniques are still relevant outside of SLAM applications. Owing to the nature of explorative navigation, the accuracy of their localisation algorithm was not considered a metric in the success of the project, but rather the performance in collision avoidance and mission execution times.



(a) Small helicopter



(b) Square landmark

Figure 2.2: Monocular pose detection of small helicopter [24].

In a thesis by Swart [24], an on-board monocular camera was used to localise a small helicopter (Figure 2.2a) during autonomous landing on a moving platform. The focus of the project was to develop a control system that could land a helicopter on the stern of a moving ship, whilst utilising a vision sensor and a landmark for localisation feedback. The landmark that was used consisted of two concentric squares (Figure 2.2b) that are large enough to be visible from various approach altitudes. Owing to the design of the landmark, several “image markers” were identified on various corners of each square. These markers were used as point landmarks to determine the relative position and

orientation of the monocular camera.

In another thesis by Malan [25], 3D tracking between satellites using monocular vision was performed. The project focus was to estimate the pose of two satellites relative to one another, travelling in formation (closer than 200 meters from each other). The satellite was modelled as a square cube with several LED markers, which could be detected as point landmarks. The satellite model was observed by an off-board, external camera (which would be mounted on another satellite in practice). The pose estimation algorithm was designed using various filtering techniques, including the extended and unscented Kalman filters. Practical tests were conducted with a high precision robotic arm to record a ground truth for the position of the satellite model. Mean pose estimation errors of 5.7 centimetres and 3.43 degrees were observed for position and orientation respectively.

Accurate vision-based localisation with manually placed landmarks is a problem that has been quite neglected in the academic field in recent years. Usually the academic focus is on SLAM related work which incorporates localisation, but the sole focus is not on the accuracy of the localisation but rather the map building. In some applications it is entirely reasonable to assume landmarks can be manually placed in the environment, which could significantly enhance the performance of a localisation algorithm that is dependant on visual landmarks. The following sections will investigate and discuss various techniques which have previously been used to implement monocular vision localisation and will assess their relevance to the problem defined in this thesis.

2.2 Estimator Algorithms

At the core of this localisation problem is the idea of estimating states from sensor data. This is done using the principle of recursive state estimation, which makes use of the knowledge of previous states and current sensor data to estimate the current states. Recursive state estimation can be implemented using some kind of filter algorithm. There are many variations and types of filters that can be used as estimators, some of which include the Kalman, extended Kalman, unscented Kalman, particle and information filters (all of which will be briefly discussed). The extended and unscented Kalman filters are some of the most commonly used in pose estimation algorithms. Motion, location and orientation estimation techniques that use Kalman filters are present in several of the papers referenced so far [22, 18, 23, 11, 24, 25]. In the case where states are represented by linear functions the general Kalman filter can be used as a Gaussian state estimator; however, in non-linear cases variations of the general Kalman filter is used. The primary purpose of both linear and non-linear applications of the Kalman filter is to perform state estimation by fusing sensor data from motion and measurement models. These sensory inputs, combined with knowledge of previously estimated states can be used to estimate the current states of the system, which may include a multi-rotor's pose, velocity or any number of other variables.

Localisation methods can be classified into absolute and relative localisation. Absolute (or global) localisation refers to a vehicle which has to localise itself relative to a global frame of reference. Relative (or local) localisation refers to state estimation in an unknown environment relative to a local object like a landmark within its vicinity. The fusion of these two methods are widely used to reduce localisation errors by employing one of the above mentioned filters, which produces a more accurate outcome than either single method alone [11]. These methods are particularly applicable to this thesis project since vision data (for relative localisation) as well as GPS data (for absolute localisation) will be available.

The Bayes filter is the most general algorithm which can be used to calculate belief distributions,

which is a probability distribution that represents the knowledge about the states given all of the available information. The Bayes filter is a recursive state estimator, which means that its current belief over certain states is calculated from the previous belief (the Bayes filter will be discussed in more detail later in this thesis). Each type of filter has specific characteristics that may make it a more suitable choice for a certain application. For the linear Kalman filter case, which conforms to the same rule-set as the Bayes filter, there are certain assumptions that have to be fulfilled. The initial belief must be Gaussian, the state transition- and measurement probability must be composed of a function that is linear in its arguments with added independent Gaussian noise. Systems that conform to these assumptions are called linear Gaussian systems. Systems that do not conform to the assumptions of linearity use other approximation techniques. Summarised here is a list of each relevant filter and its characteristics.

- The Kalman filter (KF) is the moments parametrisation case of the Bayes filter and one of the most common linear state estimators. The moments parametrisation consists of the mean (first moment) and the covariance (second moment) of the Gaussian belief distribution. Updating the Kalman filter based on a control input is computationally simple, whereas incorporating a measurement update is more difficult [26]. The normal Kalman filter is only suitable for linear systems.

- The extended Kalman filter (EKF) is a variation of the KF which is capable of estimating the belief distribution of states that are propagated through non-linear functions. A Taylor expansion is used by calculating the tangent to the non-linear function, thus making the linear filter applicable. The EKF is one of the oldest and most used filters, and in some cases it is more efficient than newer, modern variations of the KF (at the expense of precision) [27, 26].

- The unscented Kalman filter (UKF) uses a different approximation technique, called the unscented transform to account for non-linearities. It is also referred to as the derivative-free filter since it does not use the Taylor expansion method to linearise the target function, which describes the state propagation of the system through the motion or measurement models. This characteristic can be beneficial in certain systems where the derivation of functions introduce complexities. The UKF is equivalent to the KF in performance for linear systems but provides improved performance for non-linear systems. The UKF is a more modern filter and outperforms the EKF in the accuracy of its state estimations, however that may sometimes be at the expense of computational complexity [27, 26].

- The information filter (IF) is the canonical parametrisation of the Bayes filter, which consists of an information matrix and an information vector. Like the Bayes and Kalman filters, it can only be applied to linear system models, however it represents information differently than Gaussian distributions. It is simple to incorporate a measurement with the IF, however updating the filter based on a control input is difficult, which is the opposite of the KF [26, 28].

- The particle filter (PF) is a non-parametric implementation of the Bayes filter. The particle filter represents a distribution by a set of finite random samples. This representation is approximate but non-parametric and can therefore represent a much broader range of distributions than Gaussians. The particle filter also has the ability to model non-linear transformations of a random variable. However, this filter is very computationally expensive [26, 29].

The state estimation of a multi-rotor equipped with monocular vision would require a filter that can accommodate a non-linear motion model (utilising kinematic sensor data) as well as a non-linear measurement model (utilising monocular vision data). Therefore, the filters most suited to this application is the extended and unscented Kalman filters. The UKF can provide the most accurate results, whilst the EKF can be less computationally expensive in some applications; however, the state estimation will be performed offline, therefore computational complexity of either filter is not a concern. The UKF is widely used in the modern academic field of robotics and localisation, with an abundance of literature available on it. The UKF algorithm will be discussed in depth in §6.2, where its theory and implementation will be thoroughly explained.

2.3 Image Processing

In order to implement a vision-based localisation solution, it is necessary to observe a reference point in an image, referred to as a feature. The features that are observed can form a part of unique objects, such as retro-reflecting balls which are mobile, or stationary landmarks with recognisable patterns. Landmarks can take various forms, including artificially manufactured objects or natural features in the landscape, usually consisting of an object (or a part thereof) that is recognisable and unique from several viewing angles. An image frame that is taken by a vision sensor may contain several landmarks, each of which can be considered as a single measurement. The landmarks will typically occupy a small area of the image and need to be detected using some kind of image processing algorithm, which performs operations at pixel-level to determine the location of each landmark in image coordinates (illustrated in Figure 2.3). The biggest challenge in image processing applications is to determine the significance of each individual pixel and whether or not it forms part of the landmark in that frame.

In most common cases where image processing is performed, the image is converted from the red-blue-green (RBG) colour spectrum to greyscale or HSV (hue, saturation and luminance) [30]. This reduces the complexity inherent to RBG colour in pixel operations on the image but preserves a majority of the information available in the image. Each pixel therefore holds a bit of information which, in combination with the surrounding pixels and their greyscale value, presents certain recognisable features which can be used to extract measurements from each image frame. However, recognising these features is no easy task; there are a number of feature detection algorithms specifically designed to identify certain features on pixel-level in images. There is no shortage of literature available on the many types of feature detection techniques used in image processing.

One of many existing feature detection algorithms is the scale invariant feature transform (SIFT) [31]. SIFT is very robust when factoring in changes to scale and viewing angle and is very accurate as well, however the high accuracy comes at the price computational costs. Speeded-up robust features (SURF) is another feature detection algorithm that was developed as an extension to SIFT to overcome the high computational complexity of SIFT [32]. Both SIFT and SURF are feature detection algorithms that aims to detect recognisable patterns in an image that has a high probability of being recognised again in the next frame, a property called repeatability. Features include corners and blobs (a collection of pixels arranged in a specific pattern), and are usually preferred over edges and contours owing to greater accuracy in the feature's location measurement [32]. There are many other feature detection algorithms, each with its own unique characteristics, such as the Harris corner detector and features from accelerated segment test (FAST) [31]. Each one of the various algorithms are more suitable in certain applications, and can be specifically chosen to maximise the

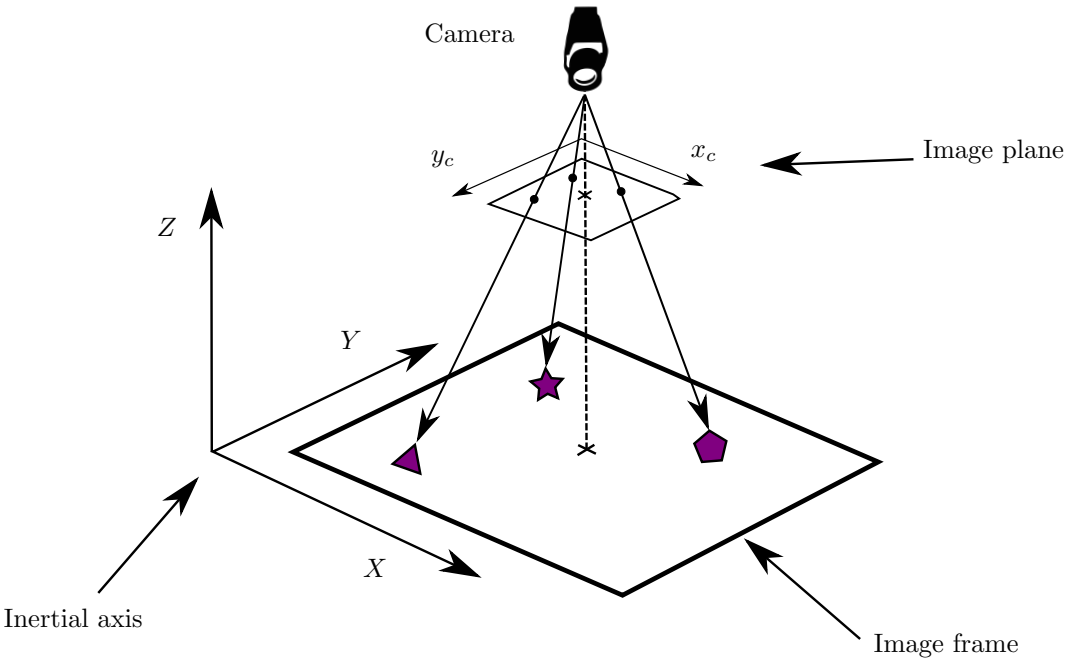


Figure 2.3: A visual illustration of how image processing can be used for localisation, where each shape is a unique landmark that falls within the image frame. Image processing is used to determine the pixel coordinates (in x_c and y_c) of each landmark in the image plane. Thereafter, with specific information about the camera hardware and landmarks, the 3D position of each landmark can be calculated in the inertial plane (XYZ). The knowledge of the 3D landmark positions in the inertial frame can be used to determine the position and orientation of the camera relative to the landmarks.

---

probability of repeatedly identifying certain features.

However, when implementing a localisation algorithm with manually placed artificial landmarks, direct implementation of image processing techniques such as SIFT, SURF or FAST presents unnecessary challenges. There are open-source packages available, such as OpenCV, which contains effective corner detection algorithms aimed at finding generic checkerboard-type landmarks in an image [33]. These open source libraries also contain implementations of existing feature detection algorithms which can be modified to identify various other types of landmarks that are dissimilar to the checkerboard style. However, there are several benefits to using a checkerboard style landmark, such as ease of use, fast detection and simple camera calibration.

Camera calibration is the process of characterising a camera (the lens and vision sensor combination) by performing image processing on several pictures of the same landmark taken from different angles. A checkerboard pattern can be described simply by the square size and number of squares in each row and column. The knowledge of the checkerboard parameters can be utilised in the process of camera calibration to yield the intrinsic camera parameters, which include information such as distortion coefficients, camera calibration matrix and focal length [34]. These parameters can be used to compensate for lens distortion and to determine the pose of the camera through the pinhole camera model.

2.4 Pinhole Camera Model

A landmark, which may consist of an object or some feature, can be identified in an image frame by image processing, which results in the known image coordinates (in pixels) of the landmark. The image coordinates of the landmark are used to calculate real-world 3D coordinates of that landmark by utilising the principles of the pinhole camera model. The pinhole camera model is an idealised representation of a camera where the aperture is described as a point and no lenses are used to focus light [35]. It describes the mathematical relationship between the coordinates of a point in 3D space and its projection on to the 2D image plane of an ideal pinhole camera. The model does not take into consideration non-ideal effects created by some lenses, such as geometric distortions or blurring of unfocussed objects. However, most of these effects can be compensated for by applying suitable coordinate transformations on the image coordinates [36]. Therefore, the pinhole camera model can often be used as a reasonable description of how a camera depicts a scene in computer vision, as described in Figure 2.4 and Figure 2.5.

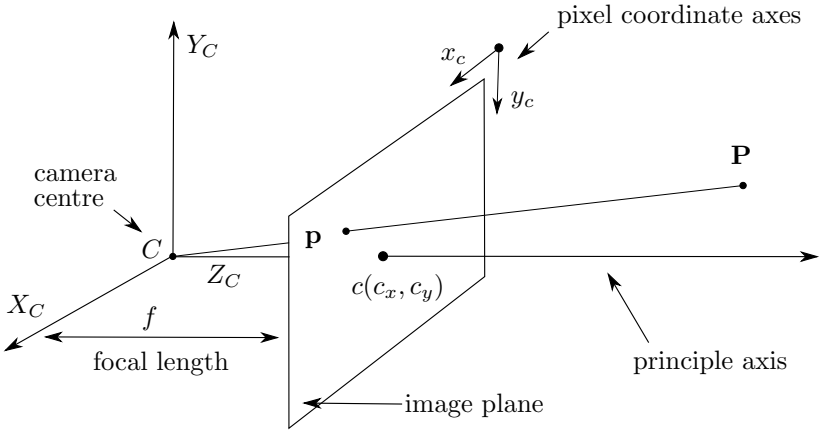


Figure 2.4: The pinhole camera model, where 'C' is the centre of the camera and the origin of the camera coordinate system ($X_C Y_C Z_C$). The image plane is perpendicular to the principle axis (Z_C) and positioned at the principle point 'c,' which is assumed to be at the centre of the image plane. Point '**P**' is a 3D point that is being observed, which is projected on to the image plane at point '**p**' and described by image coordinates (x_c and y_c in pixels). The focal length 'f' of the camera is the distance between the camera centre 'C' and the principle point.

If 3D point **P** in the camera coordinate system (in Figure 2.5) is described by

$$\mathbf{P} = \begin{bmatrix} X_P & Y_P & Z_P \end{bmatrix}^T, \tag{2.1}$$

then the image coordinates of the projected point **p** is represented by

$$\mathbf{p} = \begin{bmatrix} x_p & y_p & f \end{bmatrix}^T, \tag{2.2}$$

as indicated in Figure 2.5. If the origin of the image coordinate system is assumed to be at the principle point, then the image coordinates are calculated according to the following equations

$$\mathbf{p} = \begin{bmatrix} f(X_P/Z_P) & f(Y_P/Z_P) & f \end{bmatrix}^T, \tag{2.3}$$

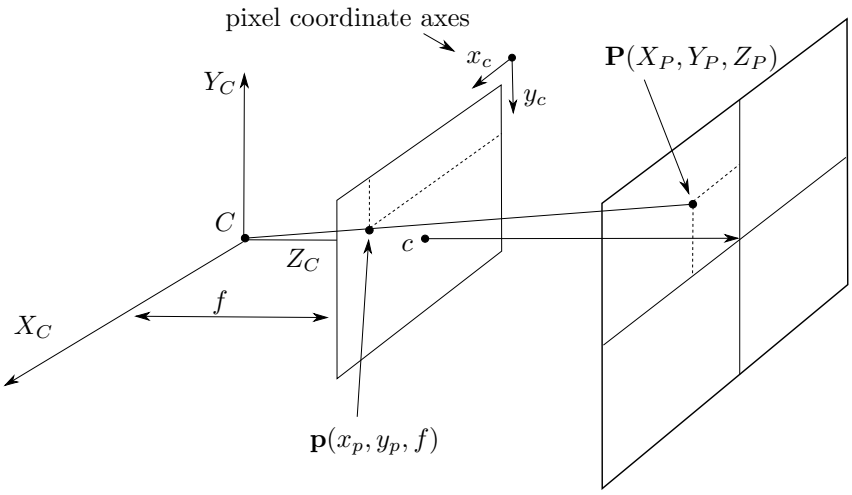


Figure 2.5: The pinhole camera model projection from '**P**' in 3D space to point '**p**' in the image plane.

---

where f is the focal length. However, to compensate for an offset of the principle point, the projection is given by

$$\mathbf{p} = \begin{bmatrix} f(X_P/Z_P) + c_x & f(Y_P/Z_P) + c_y & f \end{bmatrix}^T, \tag{2.4}$$

where c_x and c_y are the coordinates of the principle point in the image coordinate system as shown in Figure 2.4. The image point and 3D point can be described in the form of homogeneous coordinates [35]

$$\tilde{\mathbf{p}} = \begin{bmatrix} u & v & w \end{bmatrix}^T \quad ; \quad x_p = u/w \quad \text{and} \quad y_p = v/w, \tag{2.5}$$

where u, v and w are the coordinates of the projected point in pixels, and the tilde indicates that the vector is homogeneous. The homogeneous coordinates are used to describe point **P** such that

$$s\tilde{\mathbf{p}} = \mathbf{M}_c[\mathbf{R}|\mathbf{t}]\tilde{\mathbf{P}}, \tag{2.6}$$

where $\mathbf{M}_c$ is the camera calibration matrix (made up of the intrinsic parameters) and $[\mathbf{R}|\mathbf{t}]$ is a joint rotation-translation matrix, called the matrix of extrinsic parameters. The rotation-translation matrix relates the coordinates of a point to a coordinate system that is fixed with respect to the camera, such that

$$s\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{2.7}$$

where s is a scale factor for the image point, and γ is the skew parameter (which is often 0) [36].

2.5 Perspective-n-Point Problem

The perspective-three-point (P3P) or perspective-n-point (PnP) problem is used to determine the position and orientation of a camera in the world reference frame from three or more point correspondences. The problem is often used to solve for the extrinsic camera parameters ($[\mathbf{R}|\mathbf{t}]$ in Equation 2.6), and is also found in many other applications in computer vision and machine vision systems [37]. The techniques used to solve the PnP problem are directly applicable to pose estimation in this thesis, especially since vision-based sensors are used to solve the localisation problem. Given the scenario where three points are projected on to the image plane (as illustrated in Figure 2.3), the 3D positions of each respective point can be calculated by making use of the pinhole camera model and accounting for lens distortion [36]. Thereafter, the PnP problem can be solved to calculate the pose of the camera coordinate system relative to the world coordinate system wherein the points are found. There are several methods which can be used to solve the PnP problem, of which three specific solutions are implemented in OpenCV, depending on the number of point correspondences available. OpenCV's *SolvePnP* function implements the iterative, P3P and EPnP methods, of which the latter two have been published in papers [38, 39, 40].

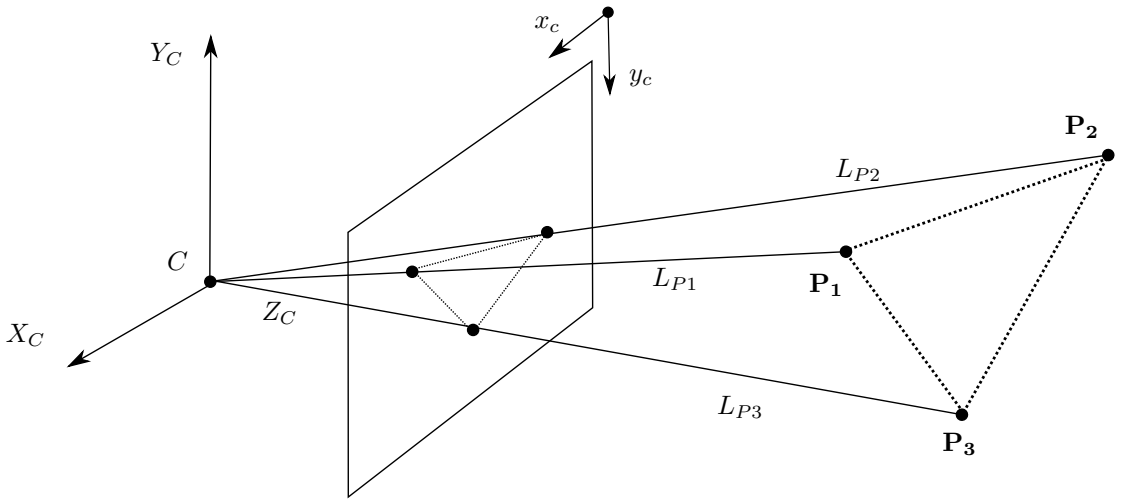


Figure 2.6: Perspective-three-point problem.

The P3P equation system is used to solve for the pose (translation and orientation) of the camera relative to three or more correspondent points. With reference to Figure 2.6, let the shortest distance between the centre of the camera and each respective point be $L_{P1} = |CP_1|$, $L_{P2} = |CP_2|$ and $L_{P3} = |CP_3|$. The angles between the correspondent points are calculated as a function of the distances between each 3D point which are projected on to the image plane, where the distances are measured in pixels. Let the angles between points be represented by $\alpha_P = \angle P_1CP_2$, $\beta_P = \angle P_2CP_3$ and $\gamma_P = \angle P_3CP_1$. The following equation system can be solved for a maximum of four finite physical solutions [38] for P_1, P_2, and P_3,

$$\left\{\begin{array}{l} L_{P1}^2 + L_{P2}^2 - L_{P1}L_{P2}2cos(\alpha) - |P_1P_2|^2 = 0 \\ L_{P2}^2 + L_{P3}^2 - L_{P2}L_{P3}2cos(\beta) - |P_2P_3|^2 = 0 \\ L_{P3}^2 + L_{P1}^2 - L_{P3}L_{P1}2cos(\gamma) - |P_3P_1|^2 = 0 \end{array}\right\}. \tag{2.8}$$

The P3P problem therefore does not provide a definite solution, since the pose of the camera relative to the correspondent points has more than one solution. However, a single solution can be found when more correspondent points are available by making use of other methods to solve the PnP problem, such as EPnP [39]. Therefore, an object, or a set of objects which are used to calculate the pose of the camera needs to collectively have more than six correspondent points that can be robustly and effectively detected by an image processing algorithm. This is another motivation for the very common checkerboard-type landmark that is used in image processing and camera calibration applications, since each corner between a black and white square is one point. If a single checkerboard-type landmark has more than six correspondent points with known positions, the pose of the camera relative to a that landmark can be calculated. Once the pose of the camera relative to the landmark is known, a single point on the landmark is chosen as a reference point to represent the measurement from the camera to that specific landmark, the application of which is discussed further in §4.3. A particularly important observation from the illustration of this problem is that a pixel-level error which may be caused by noise in the image plane would result in a constant error in the calculation of angles between points. However, the error that is made in the calculation of the shortest distance to each point from the centre of the camera is reliant on the magnitude of the distance. These observations have particular implications on the measurement noise model, which is defined in §4.3.1.

2.6 Sensor Interference

In a typical scenario where antenna equipment is flown in an RF rich environment, it is necessary to conduct a thorough interference analysis to test the robustness of sensors to external noise. The flight controller of the multi-rotor has several sensitive sensors on board, particularly an IMU which includes a magnetometer. Magnetometers are very sensitive to changes in magnetic field and prone to electromagnetic interference, which is induced by the high capacity power carrying wires that drive the multi-rotor motors. This changing of current induces a change in magnetic field which could affect the polarised test-source and other electronics that the UAV is intended to carry. This concern is addressed in Pienaar et al. [41] which discusses the use of a multi-rotor for antenna array measurements. The solution is to enclose all of the electromagnetically (EM) sensitive components in a galvanically grounded shield. This creates a more stable EM environment in which to house all of the electronics. This solution is not physically implemented in this thesis project since the focus is on the localisation algorithm in the post processing of the flight data.

Chapter 3

Data Acquisition System

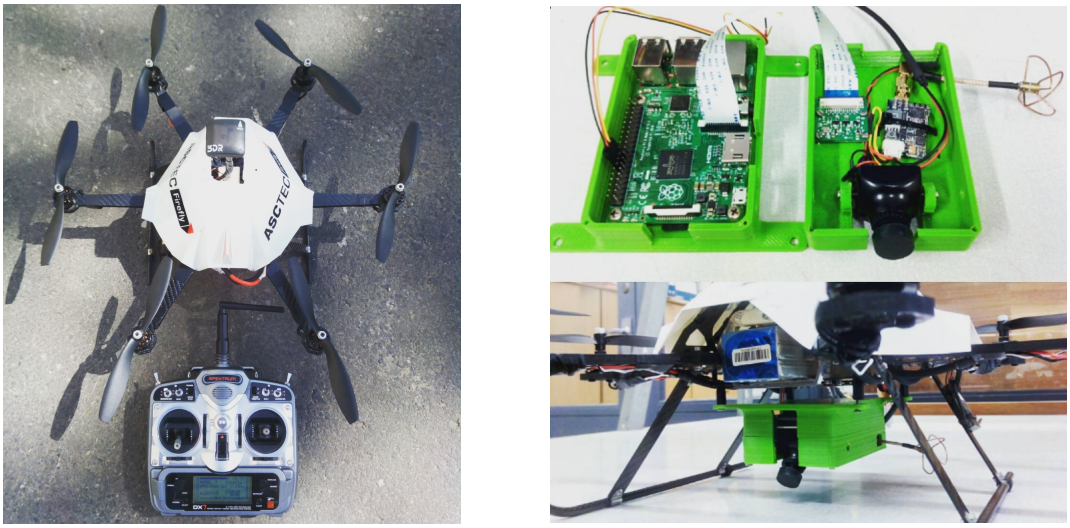
In order to perform monocular vision-based localisation, a data set is required, which may consist of images and sensor measurements. To accumulate and store data certain hardware devices and sensors are used. The hardware that is used for data acquisition is usually an important pillar on which the research topic relies and can define how the problem is approached. In this chapter the data acquisition system and relevant hardware will be introduced.

3.1 Multi-rotor Platform

The polarised test source used in the antenna characterisation process needs to be elevated to a certain altitude above the AUT. A lightweight UAV in the form of a multi-rotor is one of the simplest and least expensive solutions. For this project an Asctec Firefly hexacopter frame was used. The flight control hardware is different to the original Firefly hexacopter, only the frame and motors resemble the original multi-rotor. A 3DR Pixhawk flight controller (FC) is used to control the hexacopter, it has a multitude of integrated sensors and software which has command authority over all six motors. The multi-rotor can be manually controlled with a radio through the Pixhawk FC or it can be controlled by the Pixhawk's integrated autopilot, which can be commanded to hold a position or fly along an approximate predetermined flight path.

These features make the hexacopter very easy to operate and use for the purpose of aerial measurements. The motivation for choosing a hexacopter lies predominantly in its stability in flight and the redundancy of having six motors. If one of the motors were to fail, the hexacopter should retain enough stability to transition into a controlled descent and land, preventing damage to the payload. The Pixhawk FC has an integrated compass and an externally mounted GPS module that can be seen in Figure 3.1a. The GPS module is used to determine absolute position of the multi-rotor as well its velocity vector; however, its position measurements are only accurate to about three metres. The data acquisition hardware carried by the multi-rotor (Figure 3.1b) consists of a Raspberry Pi (RPi) computer connected to a Raspicam camera for data acquisition, as well as a CCD camera connected to a video transmitter for a live video feed. The RPi computer is also connected to the Pixhawk flight controller to capture flight information such as velocity and orientation. The entire payload is housed inside of a custom designed 3D-printed case.

The first person view (FPV) CCD camera is mounted in the payload case, connected to a 5.8 GHz video transmitter and antenna. The purpose of the FPV camera is to have a live video feed of what the image sensor is seeing, which would make the data acquisition process easier and more



(a) The Asctec Firefly

(b) The payload

Figure 3.1: (a): The Asctec Firefly hexacopter with a 3DR Pixhawk flight controller and external GPS module. The hexacopter is be manually controlled with a 2.4GHz radio. (b): The payload carried by the multi-rotor, which consists of a Raspberry Pi, Raspicam camera and a FPV camera, which is connected to a video transmitter.

reliable for the multi-rotor operator, especially at lower altitudes where the field of view of the image sensor will limit the number of visible landmarks.

3.2 Data Acquisition Hardware

The image capturing hardware is mounted below the hexacopter facing downward. The camera and data storage hardware is lightweight, small in size and easily accessible. The data storage device has to have processing capabilities in order to label each image with a timestamp and kinematic information from the Pixhawk flight controller. It is for this reason that an RPi computer is used in conjunction with a Raspicam lightweight digital camera. The Raspicam can take video or images in rapid succession, at varying resolutions and frame rates. The images are stored on a micro SD card on the RPi, which can run a script that prepares the data for off-line post-flight processing. The interaction between all the hardware devices and the flow of information is illustrated in Figure 3.2.

The RPi runs a simple version of the Linux operating system and can therefore compile and run Python scripts. The RPi communicates to the Pixhawk through the MAVLink protocol which is implemented through a Python script that is executed on the RPi. The same script monitors a trigger channel from the Pixhawk waiting for the command to start taking pictures, which can be activated from the radio that controls the multi-rotor. When it receives the command the Raspicam will start taking pictures, each picture is time-stamped and linked to a set of kinematic flight data captured at the same time as the picture. The script continues to capture data at a frequency of 5 Hz, which is limited by the frame rate of the Raspicam, until prompted to stop by the Pixhawk. Once prompted to stop, the script will package the images and flight data and transfer it on to the USB flash drive connected to the RPi.

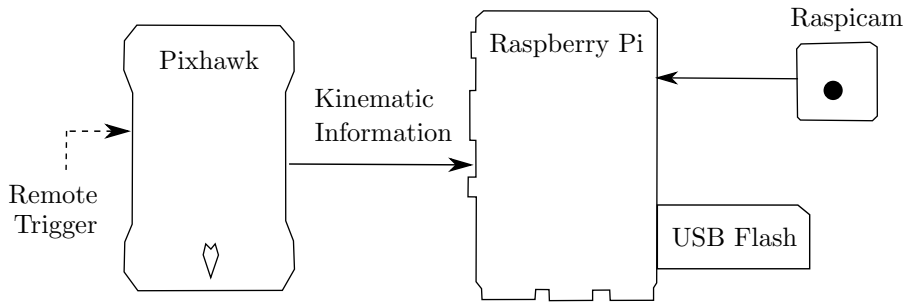


Figure 3.2: The Pixhawk controls the multi-rotor in flight and provides kinematic flight information to the RPi via UART. A remote trigger from the radio sends a signal to the RPi to start capturing images, once triggered to stop, the data is packaged and stored on the USB flash drive for post-flight processing.

3.3 On-Board Sensors

A variety of sensors are used to accumulate and capture data that can be used to perform localisation. In this section, the relevant sensors are discussed.

3.3.1 GPS module

The on-board GPS used by the Pixhawk is a consumer product based on the uBlox M8N module, which is sufficient for general flying, but not suited for accurate relative or absolute localisation. The horizontal position accuracy is 2.5 m and the accuracy of its velocity measurements is 0.05 m/s [42]. Despite the inaccuracy of the GPS module's position measurements, it can still provide relatively accurate information on the multi-rotor's velocity which can aid the localisation algorithm. The module can provide information at a frequency of up to 10 Hz, which means the maximum expected latency is 0.1 seconds [42]. The GPS module comes with a companion compass module, which has an accuracy of 0.3 degrees, and can be mounted externally and away from any noise sources (as shown in Figure 3.1a).

3.3.2 Inertial Measurement Unit

An inertial measurement unit (IMU) is used to determine the orientation and rate of rotation of the multi-rotor. The IMU used on the Pixhawk is an InvenSense MPU-6000 surface mount component which consists of a gyroscope and accelerometer [43]. Each component in the IMU provides specific information which can be used to calculated its orientation relative to the inertial axes. The gyroscope gives the rate of rotation of the IMU around its own body-fixed axes (as described in Figure 3.3) and has a programmable full scale range of 250-2000 degrees/second. The accelerometer gives the magnitude of acceleration along each axis and has a programmable full scale range of 2-16 g (where $g = 9.81$ m/s). The Pixhawk uses these measurement to calculate the multi-rotor's angle of rotation around each axis in yaw-pitch-roll Euler angles (illustrated in Figure 4.2b) by making use of an extended Kalman filter, which yields the attitude of the aircraft that the Pixhawk is integrated with.

The accuracy of the attitude estimation resulting from the EKF is highly dependent on the airframe in which the Pixhawk is integrated, since it is a flexible autopilot system. The Pixhawk

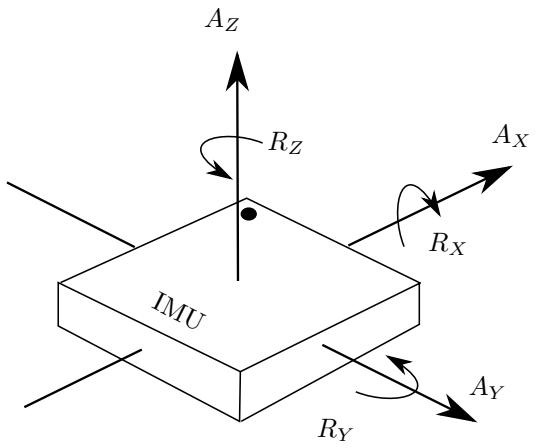


Figure 3.3: IMU axes definition (adapted from [43]), aligns exactly with the body-fixed axes definition of the multi-rotor in Figure 4.2b.

supports a multitude of airframes including fixed-wings and multi-rotors, each of which induce noise in different ways. The accuracy of the attitude estimation from the Pixhawk is also dependent on the way that the flight controller is mounted to the aircraft and whether or not vibration dampeners are used. In this application, the orientation of the multi-rotor is directly measured from the attitude estimation by the Pixhawk, therefore the noise in the measurements is characterised for this specific hardware set-up and will be discussed further in §4.2.1.

3.3.3 Image Sensor

The Raspicam is an inexpensive digital image sensor designed for the RPi to allow easy integration and use. The module that was used in this thesis project is revision 1.3 of the module (pictured in Figure 3.4), and the accompanying hardware and software utilities makes it a very good prototyping tool for computer vision applications. The camera has a five megapixel OmniVision OV5647 sensor, with a maximum resolution of 2592 x 1944 pixels and a focal length of 3.6 millimetres. The sensor image area is 3.76 x 2.74 millimetres, and the size of each pixel is 1.4 micrometres. The camera implements a rolling shutter and has a frame rate up to 47 frames per second, depending on the resolution and software features that are used [44].

The lens used in the Raspicam is the full-frame equivalent of a 35 millimetre SLR lens, with a physical size of 6.35 millimetres. The lens provides one metre to infinity focus and a field of view of 53 and 41 degrees in the horizontal and vertical directions respectively. The relatively narrow field of view of the Raspicam has some advantages in image processing: there is very little distortion in the images due to small lens curvature, and the loss in measurement accuracy when pictures are taken at an angle relative to the landmark is negligibly small. This means that there is no stable flight condition of the multi-rotor where the pitch and roll angles would be so extreme that the sensor measurements taken by the camera would inaccurately represent objects in the image.

The Raspicam is a hobbyist product and boasts no exceptional features that are found in expensive specialised computer vision cameras. However, when a localisation problem is approached with a vision-based solution, it is necessary to consider the effect of image resolution on the pose estimation accuracy that results from pixel-level image processing operations. The measurement noise

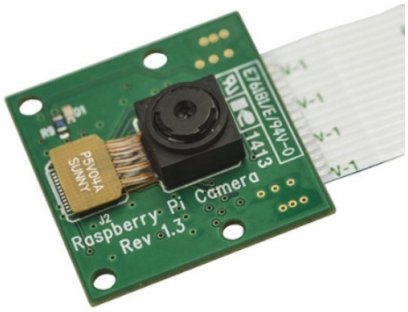


Figure 3.4: The Raspicam camera board module [45]

in pixel coordinates is expected to be relatively constant owing to the constant sensor noise in the image plane. Furthermore, the amount of sensor noise present in each measurement is expected to be directly proportional to the size of each pixel. Therefore, by doubling the resolution of the image sensor (or halving the pixel size) whilst keeping the field of view constant, the standard deviation in sensor noise in the image plane should decrease by half. Thus, the accuracy in 3D localisation will be higher since the pixel measurements of the 3D projection on the image plane will only be subject to half as much noise than a lower resolution sensor. The magnitude of the impact that a higher resolution image sensor will have on the accuracy of the localisation algorithm is challenging to predict without a higher resolution image sensor for comparison and exact measurement and sensor noise values. However, it is not unreasonable to confidently assume that less sensor noise will result in more accurate image measurements.

A better resolution sensor, along with other hardware improvements will undoubtedly result more accurate state estimation. However, despite the benefits of using advanced and expensive high resolution image sensors, the goal of this project is to use inexpensive hardware and analyse the resulting localisation accuracy to determine if it is a sufficient solution for antenna characterisation.

3.3.4 Baseline DGPS

For final testing of the localisation algorithm, an absolute baseline is required with which to compare the accuracy of the localisation algorithm developed in this thesis. The ideal baseline measurements would have millimetre accuracy, which is achievable by systems such as the Vicon motion capture system. However, the most accurate solution that was available during this thesis project was Piksi differential GPS system (DGPS). The DGPS utilises two GPS modules, one which is stationary (called the base) and one which is mobile (called the rover). A good DGPS system can perform absolute localisation at up to one centimetre accuracy, however the high accuracy comes at an extremely high price. Lately, however, there have been developments by smaller companies to develop inexpensive DGPS systems aimed at the hobbyist consumer market. One of these systems is the Swift Piksi GNSS module, which is capable of two to five centimetre accuracy under ideal conditions [46], at a significantly lower cost. A Piksi DGPS system was available at the time of this project, and will therefore be used to measure a baseline against which to compare the localisation algorithm's performance.

3.4 Summary

The data acquisition hardware was introduced, which consists of a multi-rotor hexacopter platform with a Pixhawk flight controller. The flight controller module includes several sensors, such as an IMU, GPS and compass, which can provide a plethora of flight information. Furthermore, a payload is mounted underneath the multi-rotor which includes a Raspberry Pi on-board computer and a Raspicam camera that is used to take image measurements. The importance of image resolution is discussed with special consideration of the goals of this project. Lastly, the Piksi differential GPS system is introduced, which is used to measure the ground truth position of the multi-rotor during experimental flight tests.

Chapter 4

System Modelling

The nature of any localisation problem is such that it requires a description of the system it applies to. The description should encompass a realistic model of all the state variables which are relevant to the estimation problem. This includes how the system moves in space (called the motion model) as well as how these movements affect the measurements from the various sensors (called the measurement model). No real-world system can be perfectly described, there is often inaccuracies in the approximations and assumptions that are made to realise a system model mathematically. Furthermore, the data that is obtained from sensor measurements are often plagued with noise, which needs to be characterised by constructing noise models. An accurate description of the relationship between the states, noise models, the motion model and the measurement model will enable the UKF to perform its function as a state estimator. This chapter aims to define the system model and describe each subsystem sufficiently.

4.1 System Kinematics

The kinematics of a system describes the state and motion of the system by directly sampling the values of the relevant sensors without derivation from the forces (or commands) that caused those motions to take place. Whereas kinetics refers to the forces (initiated by commands or resulting from a different process) that cause motions which induce certain measurement values in the relevant sensors. The modelling of any system can quickly become extremely complex, especially when control inputs, external and internal forces as well as measurements need to be described. The kinetic modelling of a multi-rotor is very complex, especially when accurate control of the system is required. However, when a system's motion and states are only observed, the system model is simplified significantly. The states are related to the sensors through mathematical equations in both the motion and measurement models. It is possible to also consider the control input that attempts to manipulate states through kinetics, however this would significantly complicate the motion model of the multi-rotor. Furthermore, owing to the nature of multi-rotor aircraft and their ability to easily manoeuvre in six degrees of freedom, there is no guarantee that a certain control input will cause a specific state, owing to external factors that may have an adverse effect (such as wind). This thesis focuses on the accurate observation and estimation of system states realised through kinematics equation; therefore, only sensor values are measured and the control inputs are not considered.

4.1.1 State Variables

The pose of the multi-rotor refers to its position and orientation, which can be represented by six states. A position in 3D space relative to the landmark coordinate system can be described by the following state variables: X_t, Y_t and Z_t, which gives the magnitude and direction of displacement along each relevant axis. These three state variables provide the position of the multi-rotor relative to some predetermined point of origin in an inertial coordinate system. The point of origin in this thesis is defined as the origin of the axes along which the landmarks are placed (hereafter referred to as the landmark axes), as shown in Figure 4.2a.

The orientation of the multi-rotor in the landmark reference frame can be informally described by the following three state variables: ϕ which represents roll , θ which represents pitch and ψ which represents yaw. All of the rotations are positive in the counter-clockwise direction according to the right-hand-rule. The exact orientation of the multi-rotor is described in full by a set of Euler angles, which is subject to an order of hierarchy. In this thesis the yaw-pitch-roll Euler angle convention is used, which is discussed in §4.1.2. A zero pitch and roll angle will result in the multi-rotor being perfectly level and parallel with the X-Y plane of the landmark axes. A zero yaw angle will align the front of the multi-rotor (the body-fixed x-axis) with the landmark Y-axis. The body-fixed axes of the multi-rotor is defined in Figure 4.2b.

Owing to the yaw-pitch-roll Euler angle convention, the yaw angle of the multi-rotor in the landmark axes is not equivalent to the heading angle which is measured by the compass. The multi-rotor's yaw angle and velocity vector is measured in a different inertial frame of reference as the landmark axes. However, by making use of the compass, the velocity vector provided by the GPS, and landmark measurements, it is possible to perform axes transformations that describe all of the sensor measurements in the landmark axes. Therefore a seventh state is introduced: a constant state which describes a yaw angle in the landmark reference frame, positive in the counter-clockwise direction around the landmark Z-axis according to the right hand rule. This state is called the landmark alignment angle, and describes the difference between the north-axis in the inertial north-east-down (NED) reference frame (wherein the velocity vector is given) and the landmark Y-axis. All of the state variables are summarised in Table 4.1.

Table 4.1: All of the state variables, their units and symbols

State Name	Symbol	Unit	Description
X-Position	X_t	metres	Position along the landmark X-axis
Y-Position	Y_t	metres	Position along the landmark Y-axis
Z-Position	Z_t	metres	Position along the landmark Z-axis
Yaw	ψ_t	degrees	Euler angle in landmark reference frame
Pitch	θ_t	degrees	Euler angle in landmark reference frame
Roll	ϕ_t	degrees	Euler angle in landmark reference frame
Landmark-alignment	O_{LA}	degrees	Angle between landmark Y-axis and NED North-axis

The state variables at the current time step are represented by the state vector $\mathbf{x}_t$

$$\mathbf{x}_t = \begin{bmatrix} X_t & Y_t & Z_t & \psi_t & \theta_t & \phi_t & O_{LA} \end{bmatrix}^T. \tag{4.1}$$

4.1.2 Euler Angle Transformations

In this thesis, the yaw-pitch-roll (also known as 3-2-1) Euler angle convention is used, which means that any given vector in the multi-rotor body-fixed reference frame can be described in the inertial landmark reference frame by first performing the roll-angle, then the pitch-angle and thereafter the yaw-angle rotation, where each rotation takes place in the rotated body-fixed axes system (as illustrated in Figure 4.1).

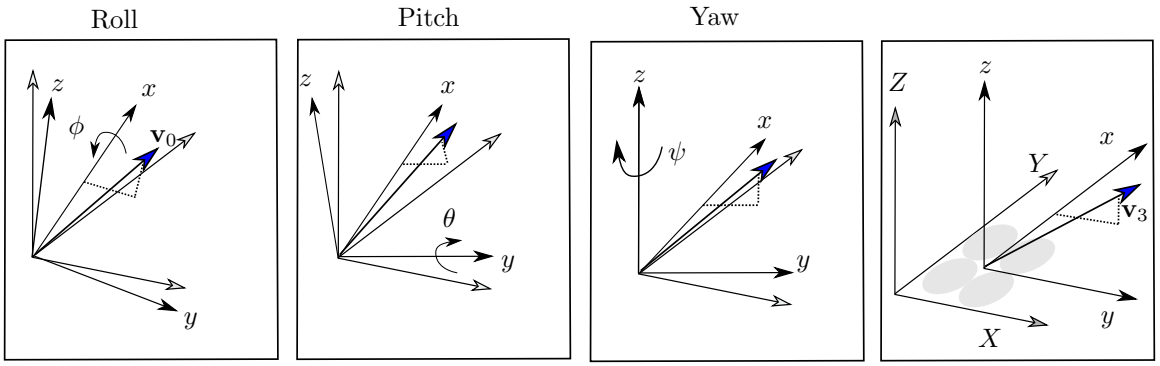


Figure 4.1: A stepwise illustration of three Euler angle transformations on a vector in the multi-rotor body-fixed axes (xyz) with respect to the inertial landmark reference frame (XYZ).

In traditional aircraft dynamics, the front of the aircraft is denoted by the x-axis in the body-fixed reference frame [47], the same convention is used in this thesis, where the front of the multi-rotor is denoted by the x-axis in the body-fixed reference frame (see Figure 4.2b). In order to describe a vector that is in the body-fixed axes of the multi-rotor in a different inertial reference frame, the first rotation that needs to take place is roll, and it is mathematically performed using a rotation matrix. If $\mathbf{v}_0$ is a vector that can be described by coordinates in the body-fixed axes,

$$\mathbf{v}_0 = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T, \tag{4.2}$$

then a roll rotation can be performed around the body-fixed x-axis with the rotation matrix $\mathbf{R}_{(\phi)}$,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \text{-sin}(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}. \tag{4.3}$$

Thereafter, a pitch rotation is performed around the new body-fixed y-axis with another rotation matrix $\mathbf{R}_{(\theta)}$,

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ \text{-sin}(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{4.4}$$

Lastly, a yaw angle rotation is performed around the new body-fixed z-axis with a final rotation matrix $\mathbf{R}_{(\psi)}$,

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \text{-}\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi & 0) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}. \tag{4.5}$$

The original vector $\mathbf{v}_0$ is now described in the inertial landmark reference frame by

$$\mathbf{v}_3 = \begin{bmatrix} x_3 & y_3 & z_3 \end{bmatrix}^T. \tag{4.6}$$

The three rotation matrices that describe the Euler angle transformations can be multiplied for simplicity, such that

$$\mathbf{v}_L = \mathbf{R}_{(\psi)}\mathbf{R}_{(\theta)}\mathbf{R}_{(\phi)}\mathbf{v}_B, \tag{4.7}$$

where $_L$ and $_B$ denote the landmark and body-fixed axes, and $\mathbf{R}$ denotes each respective rotation matrix. Furthermore, the same transformation can be reversed to describe a vector in the inertial landmark axes with respect to the body-fixed reference frame,

$$\mathbf{v}_B = \mathbf{R}^{-1}_{(\phi)}\mathbf{R}^{-1}_{(\theta)}\mathbf{R}^{-1}_{(\psi)}\mathbf{v}_L, \tag{4.8}$$

which reverses the order of the matrix operations and the direction of rotation.

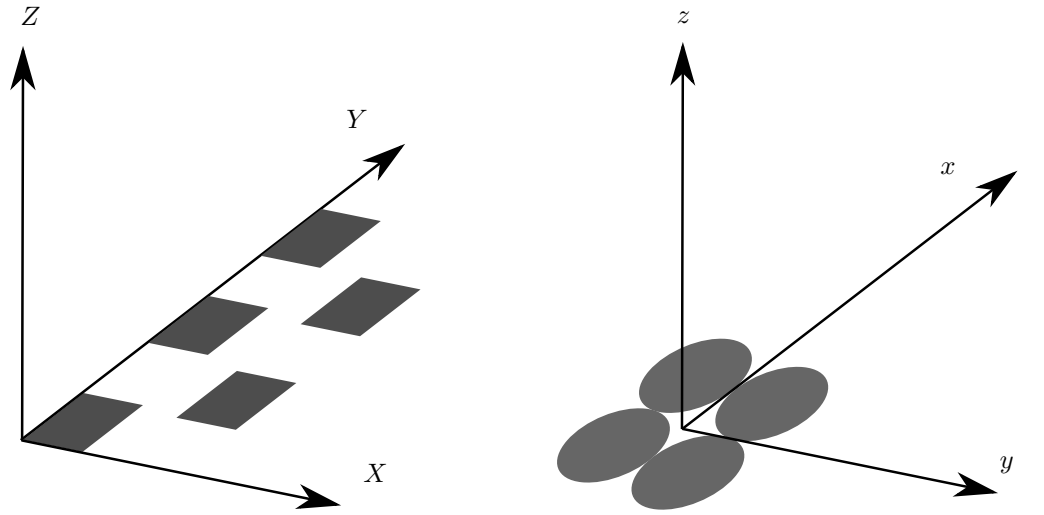
4.1.3 Axes Transformations

The XYZ position coordinates describe the relative displacement of the multi-rotor in the inertial landmark axes, which is different from the inertial reference frame wherein some measurements are defined. Some of the sensors on the multi-rotor, like the GPS, gives kinematic information (such as velocity) in a coordinate system called the north-east-down (NED) reference frame. The inertial NED axes is defined at the position where the multi-rotor takes off, where the north- and east-axis aligns with the Earth's geographical northern and eastern axes, and the down-axis points to the centre of the Earth. It is therefore necessary to find a relationship between the landmark axes and the NED axes. The same applies to the Euler angles of the multi-rotor, which describe the body-fixed axes of the multi-rotor with respect to the NED reference frame and needs to be transformed to the landmark reference frame. The desired end result of all the rotation transformations is to have all of the states and measurements in the same inertial axes system in order to perform successful localisation.

The axes transformations are performed using either Equation 4.7 or 4.8, depending on the direction of the rotation transformation (from the inertial landmark axes to another reference frame, or vice versa). The product of the three rotation matrices in Equations 4.3 - 4.5 is often referred to as the direction cosine matrix (DCM) [48], which is only intended to simplify the three matrices by replacing it with one.

NED to Landmark Axes

An important axes transformation that makes use of the DCM is performed on velocity sensor data obtained from the multi-rotor. This axes transformation aims to relate the velocity measurements from the Pixhawk to the landmark axes system. The velocity measurements are presented in a vector format which is described in the inertial NED reference frame. Since the landmarks are placed flat on the surface of the Earth, the X-Y plane in the landmark axes is parallel to the north-east plane,



(a) Landmark axes definition

(b) Multi-rotor body-fixed axes definition

Figure 4.2: (a) The landmark axes definition where the X-Y plane is parallel to the earth's north-east plane, and the Z-axis is altitude above the landmarks, which are placed in the X-Y plane. (b) The body-fixed axes definition, where the x-axis denotes the front of the multi-rotor.

and the Z-direction in the landmark axes is aligned with down in the NED axes. However, a positive Z-displacement in the landmark axes relates to a negative down displacement in the NED axes, this is accounted for by rolling the velocity vector by 180 degrees around its north-axis (step 1 in Figure 4.3). The landmark axes system has a constant yaw angle difference between its Y-axis and the north-axis, which is estimated and given by the landmark alignment angle. This is accounted for by rotating the velocity vector around its down-axis (which now faces up) by a magnitude of the landmark alignment angle in order to align the north-axis with the Y-axis (shown in step 2). The rotation matrices are therefore used to relate a velocity vector in the NED axes from the Pixhawk to a velocity vector in the landmark axes where north aligns with Y, east aligns with X and down aligns with Z. The axes transformation is visually illustrated in Figure 4.3, where the dotted lines are the landmark axes as illustrated in Figure 4.2a.

The rotation transformation that is performed in Figure 4.3 uses a DCM, which is shown in Equation 4.9, where ϕ_v, θ_v and ψ_v are the specific roll, pitch and yaw angles, the $_{XYZ}$ subscripts refers to the landmark axes and the $_{NED}$ subscripts refers to the inertial NED axes. The full DCM equation [47] is

$$\begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} = \begin{bmatrix} C_{\psi_v}C_{\theta_v} & C_{\psi_v}S_{\theta_v}S_{\phi_v} - S_{\psi_v}C_{\phi_v} & C_{\psi_v}S_{\theta_v}C_{\phi_v} + S_{\psi_v}S_{\phi_v} \\ S_{\psi_v}C_{\theta_v} & S_{\psi_v}S_{\theta_v}S_{\phi_v} + C_{\psi_v}C_{\phi_v} & S_{\psi_v}S_{\theta_v}C_{\phi_v} - C_{\psi_v}S_{\phi_v} \\ -S_{\theta_v} & C_{\theta_v}S_{\phi_v} & C_{\theta_v}C_{\phi_v} \end{bmatrix} \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \quad C_{()} = \cos(), S_{()} = \sin(), \tag{4.9}$$

where the following variable values apply for the velocity vector transformation from NED axes to landmark axes:

$$\phi_v = 180 \text{ deg}, \quad \theta_v = 0 \quad \text{and} \quad \psi_v = O_{LA}. \tag{4.10}$$

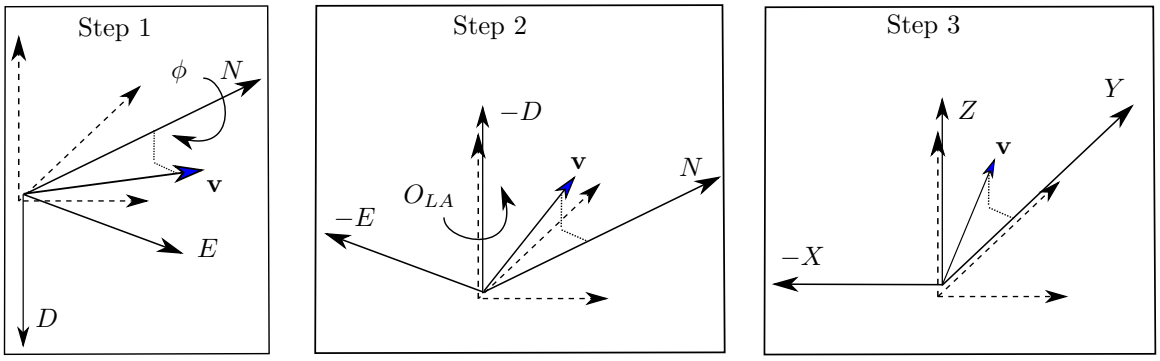


Figure 4.3: A stepwise illustration of the transformation of the velocity vector from the NED axes to the landmark axes.

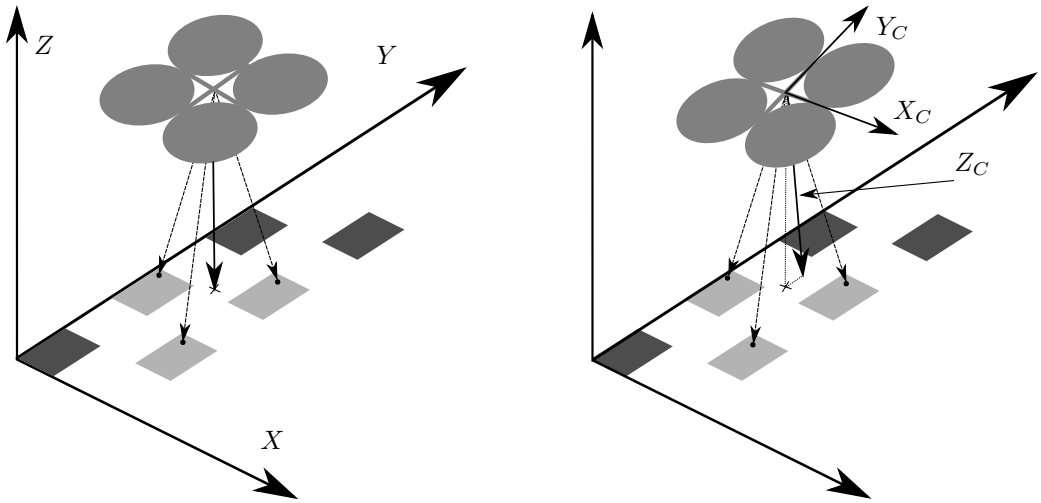
Landmark to Body-fixed Axes

Another important axes transformation is on a vector which points from the centre of the camera to a landmark. The transformation relates the vector in the landmark reference frame (Figure 4.4a) to the body-fixed reference frame (Figure 4.4b). The purpose of this axes transformation is to enable an accurate prediction of what the landmark measurements look like from the perspective of the camera (the necessity for this prediction is motived and discussed in §4.3). The camera reference frame (shown in Figure 4.4b) is aligned with the multi-rotor body-fixed axes, such that x and y aligns with Y_C and X_C respectively, and Z_C aligns with z in the negative direction. In the scenario where the multi-rotor is perfectly level (all angles of rotation are zero, depicted in Figure 4.4a), the prediction of the landmark measurement vectors would match perfectly (not considering sensor noise) with the measured positions which is calculated through image processing. However, if the camera (which is rigidly mounted to the multi-rotor) is oriented by any degree around the body-fixed roll, pitch or yaw axes (as in Figure 4.4b), the measured position vectors of the landmarks would differ from the predicted measurement vectors. This can be accounted for by performing the axes transformation described in Equation 4.8, which will calculate the orientation of a vector in the landmark reference frame (from the camera to some landmark) in relation to the rotated body-fixed reference frame to predict how the landmark measurements look from the perspective of the camera.

It can be shown that the three dimensional Euler rotation matrix is orthogonal, which means its inverse is equal to its transpose [47], therefore Equation 4.8 can also be written as

$$\mathbf{v}_B = \mathbf{R}^T_{(\phi)}\mathbf{R}^T_{(\theta)}\mathbf{R}^T_{(\psi)}\mathbf{v}_L, \tag{4.11}$$

where $\mathbf{v}_L$ is a vector (protruding from the centre of the camera) in the landmark reference frame, and $\mathbf{v}_B$ is the same vector described in the body-fixed (or camera) reference frame. The DCM resulting from the product of the rotation matrices is



(a) Measurement vectors in landmark axes

(b) Measurement vectors in body-fixed axes

Figure 4.4: (a): The landmark measurement vectors in the landmark axes, where the multi-rotor is perfectly level. (b): The same landmark measurement vectors, however described in the camera (or body-fixed) axes of the multi-rotor which is orientated by some degree relative to the landmark reference frame.

$$\begin{bmatrix} v_{Bx} \\ v_{By} \\ v_{Bz} \end{bmatrix} = \begin{bmatrix} C_{\psi_B}C_{\theta_B} & S_{\psi_B}C_{\theta_B} & -S_{\theta_B} \\ C_{\psi_B}S_{\theta_B}S_{\phi_B} - S_{\psi_B}C_{\phi_B} & S_{\psi_B}S_{\theta_B}S_{\phi_B} + C_{\psi_B}C_{\phi_B} & C_{\theta_B}S_{\phi_B} \\ C_{\psi_B}S_{\theta_B}C_{\phi_B} + S_{\psi_B}S_{\phi_B} & S_{\psi_B}S_{\theta_B}C_{\phi_B} - C_{\psi_B}S_{\phi_B} & C_{\theta_B}C_{\phi_B} \end{bmatrix} \begin{bmatrix} v_{Lx} \\ v_{Ly} \\ v_{Lz} \end{bmatrix} \quad C_{()} = \cos(), S_{()} = \sin(), \tag{4.12}$$

where ϕ_B, θ_B and ψ_B are the roll, pitch and yaw Euler angles of the multi-rotor.

4.2 Motion Model

The motion model in this thesis describes how the multi-rotor's states change from one time step to the next, given the previous states and current kinematic sensor information. If $\mathbf{x_p}$ is a vector which holds the X, Y and Z position states of the multi-rotor, δ is the change in time and $\mathbf{v}_{t,L}$ is a vector that holds the current velocity in the landmark XYZ-axes, then

$$\mathbf{x}_{\mathbf{p}t} = \mathbf{x}_{\mathbf{p}t-1} + \delta(\mathbf{v}_{t,L} + \mathbf{q}_v), \tag{4.13}$$

where $X_t = \mathbf{x}_{\mathbf{p}_t}[0]$, $Y_t = \mathbf{x}_{\mathbf{p}_t}[1]$ and $Z_t = \mathbf{x}_{\mathbf{p}_t}[2]$. Equation 4.13 therefore approximates how the position of the multi-rotor would change, given the position $\mathbf{x}_{\mathbf{p}_{t-1}}$ at the previous time step and velocity sensor measurements with additive sensor noise $\mathbf{q}_v$. This state update equation is only valid for small increments of δ where the velocity measurement remains constant between time steps. This approximation is valid in this thesis project, since test flights are conducted in a slow and smooth manner whilst sampling data at a rate of 5 Hz such that $\delta <= 0.2$ s. The rate of change in altitude (velocity in the landmark Z-axis) of the multi-rotor is calculated by an estimator which makes use of a barometer, whereas the horizontal velocity is calculated from an estimator which uses GPS and

inertial sensors. Therefore, none of the sensors used to estimate the velocity of the multi-rotor are sensitive enough to measurement variations in velocity within the δ time step.

The orientation of the multi-rotor is measured directly from the Pixhawk's attitude estimator, which makes use of IMU measurements to calculate the Euler angles in the NED reference frame. However, the yaw value (ψ_t) is adjusted to align the NED axes to the landmark axes by making use of the landmark alignment angle. The orientation is therefore

$$\phi_t = u_{t,\phi} + q_\phi, \qquad \theta_t = u_{t,\theta} + q_\theta \quad \text{and} \quad \psi_t = u_{t,\psi} - O_{LA} + q_\psi, \tag{4.14}$$

where $\boldsymbol{u}_t$ are the current orientation measurements, $\boldsymbol{q}$ is the relevant measurement noise and O_{LA} is the landmark alignment angle. Therefore all six of the current state values can be updated according to

$$\mathbf{x}_t = \boldsymbol{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t, \tag{4.15}$$

where $\boldsymbol{g}$ represents all of the motion model update equations, and $\mathbf{w}_t$ is the additive process noise.

4.2.1 Process Noise

Any measurement provided by a sensor is subject to some form of noise, especially if less expensive hardware components are used. Since it is not possible to rely on perfect sensors it is necessary to model the noise that these sensors produce in their measurements. By modelling the sensor noise for the motion model (referred to as process noise), the system model is made robust to a certain margin of error in sensor measurement values. In this application, additive Gaussian process noise is introduced to the motion model's state propagation. The process noise was calculated by taking many sensor measurements over a certain time period whilst keeping the system stationary. The immobile system should not induce any changes in sensor measurements such as velocity and orientation, and therefore any variation in the measurements is considered noise, which is represented by a certain variance and a mean of zero. However, the noise that is present in the sensor measurements may vary, depending on the operational state of the multi-rotor. Furthermore, the spinning motors induce magnetic fields by rapidly changing currents, which may also have unknown effects on the sensor noise. The latter scenario was simulated by rigidly fastening the multi-rotor to a platform whilst spinning the motors, during which time the sensor measurements were taken. The resulting data set should sufficiently characterise the sensor measurement noise under operational conditions during flight tests. The noise variance for each relevant sensor was calculated from the data set resulting from the stationary test,

$$\sigma^2 = \sum_{i=0}^{n} \left((s_i - \bar{s})^2 / (n-1) \right), \tag{4.16}$$

where $\bar{s}$ is the average of the data set, s_i is a specific value in the data set and n is the number of samples. However, the process noise should describe the expected noise with regards to each state, which is not the case for the velocity noise measurements, which are used to update the position states in the motion model. The noise variables related to the X_t, Y_t and Z_t position states are therefore updated for each prediction step,

$$\sigma_{\mathbf{x}_t} = \sigma_{\mathbf{v}_t} \delta, \tag{4.17}$$

where $\boldsymbol{\sigma}_{\mathbf{x}_t}$ is a vector containing standard deviations for the velocity noise in each relevant axis, such that

$$\sigma_X^2 = \sigma_{\mathbf{x}_t}[0]^2 \quad , \quad \sigma_Y^2 = \sigma_{\mathbf{x}_t}[1]^2 \quad \text{and} \quad \sigma_Z^2 = \sigma_{\mathbf{x}_t}[2]^2. \tag{4.18}$$

The noise variance for the orientation states remain constant. The noise value that is additively introduced to each state in the system is drawn from the normal distribution, which has a certain variance and a mean of zero,

$$q_n \sim \mathcal{N}(0, \sigma_n^2), \tag{4.19}$$

where q_n is the relevant noise value to be added to the state, and σ_n^2 is the noise variance. All of the noise variances are represented by the diagonal process noise covariance matrix $\mathbf{Q}_p$

$$\mathbf{Q}_p = \begin{bmatrix} \sigma_X^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_Y^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_Z^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\psi^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{O_{LA}}^2 \end{bmatrix}, \tag{4.20}$$

where σ^2 is the noise variance of the relevant state variable. The noise is additively introduced into the system in the UKF motion model update (prediction step), according to Equation 4.15, where the noise values are drawn from the normal distribution described by the process noise covariance matrix

$$\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_p). \tag{4.21}$$

In this thesis, the process noise is is assumed to be independent, which is why it is represented by a diagonal matrix where all of the off-diagonal values are zero. However, in reality, the process noise that is produced by the various sensors is most probably dependent. The sensor measurements (and therefore the sensor noise) is the output of an estimator on the Pixhawk, which makes it impossible to explicitly measure the covariance between different noise processes. Since all of the measurements produced by the Pixhawk are the results of an estimator, it is very difficult to measure the exact dependent process noise matrix used in the measurement estimation.

The practical implications of not knowing the exact values of the entire process noise matrix may be marginally less accurate state estimates after the prediction step. The only alternative is to process raw measurements from the gyroscope, accelerometer and GPS and design a filter which estimates the relevant measurement values. However, that would require a lot of time and unnecessary effort, whilst bearing very few benefits. Furthermore, the measurement model and measurement noise is expected to have a much larger impact on the accuracy of the localisation algorithm.

4.3 Measurement Model

The purpose of the measurement model is to describe how the landmark measurements look from the image sensor given the states and parameters of the system. The measurement model therefore

describes the mathematical relationship between the states of the multi-rotor and the measurements taken by the vision sensor. This information is used to predict what the image sensor measurements will be for a certain state vector, which can then be compared to the actual measurements to aid the localisation algorithm in its state estimation. Each image taken by the camera may hold several measurements, where each visible landmark is considered as one measurement. The image processing algorithm is designed to calculate the 3D position of each landmark from the perspective of the camera in the camera reference frame. The $X_C Y_C Z_C$ coordinates of each landmark in the camera axes is aligned with the body-fixed axes as illustrated in Figure 4.5b, and can therefore easily be described in xyz body-fixed coordinates. The Cartesian coordinates of each landmark in the body-fixed axes is converted to spherical coordinates in range (R), azimuth (α) and elevation (β) values. These spherical coordinates give a simple description of the 3D position of the landmarks relative to the camera in the body-fixed axes of the multi-rotor. Using spherical coordinates for landmark positions provides the most static and simple description of noise in the image measurements.

In order to predict what the landmark measurements will look like from the perspective of the camera, it is necessary to transform the landmark measurement vector in the landmark axes (Figure 4.5a) to a vector that is described in the body-fixed axes (Figure 4.5b). It is therefore necessary to determine the position of the multi-rotor relative to the relevant landmark in the landmark axes, which can be calculated using the current XYZ states and the known landmark position coordinates (X_L, Y_L and Z_L), such that

$$\mathbf{v}_L = \begin{bmatrix} x_L & y_L & z_L \end{bmatrix}^T, \tag{4.22}$$

where $x_L = X_t - X_L$, $y_L = Y_t - Y_L$ and $z_L = Z_t - Z_L$ (indicated in Figure 4.5a). The vector that describes the landmark position relative to the multi-rotor in the landmark reference frame is then related to the body-fixed reference frame through the axes transformation in Equation 4.23,

$$\mathbf{v}_B = \mathbf{R}^T_{(\phi_B)} \mathbf{R}^T_{(\theta_B)} \mathbf{R}^T_{(\psi_B)} \mathbf{v}_L, \tag{4.23}$$

where ϕ_B, θ_B and ψ_B denotes the orientation states of the multi-rotor relative to the landmark reference frame. Therefore, the position of the landmark in the body-fixed reference frame is denoted by $\mathbf{v}_B$, such that

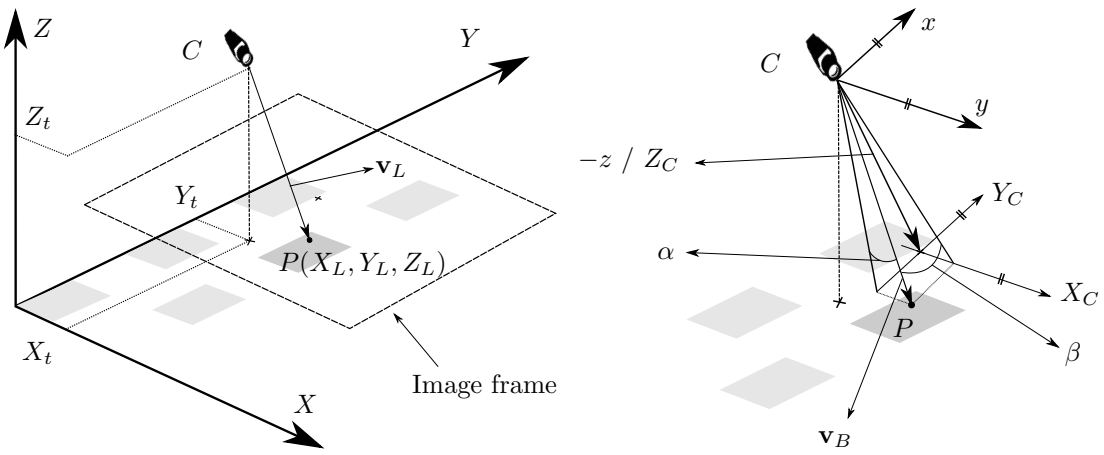
$$\mathbf{v}_B = \begin{bmatrix} x_B & y_B & z_B \end{bmatrix}^T. \tag{4.24}$$

Thereafter, the vector is converted to spherical coordinates in range, azimuth and elevation, which describe the relative position of the landmark in the body-fixed axes. The calculations used to determine the spherical coordinates are shown in Equation 4.25 - 4.27, where the range (R) is the shortest distance between the centre of the camera and the landmark point measurement,

$$R = \sqrt{(x_B)^2 + (y_B)^2 + (z_B)^2} + q_R, \tag{4.25}$$

and q_R is additive measurement noise. The azimuth (α) is the angle between the landmark point measurement and the body-fixed x-axis (or camera Y_C-axis), as viewed from the perspective of the camera,

$$\alpha = \pi/2 - \arctan((z_B)/(-y_B)) + q_\alpha, \tag{4.26}$$



(a) Measurement vector in landmark axes

(b) Measurement vector in body-fixed axes

Figure 4.5: (a): A landmark measurement vector $\mathbf{v}_L$ for a single measurement P, as calculated in the landmark reference frame. However, three landmark measurements are visible in the image frame, so this vector will exist for each visible landmark in practice. (b): The landmark measurement vector $\mathbf{v}_B$ transformed to the body-fixed reference frame, where the range is $R = |\mathbf{v}_B|$, azimuth is α and elevation is β. The camera $X_C Y_C$ and body-fixed xy axes are drawn as a parallel projection of one another for illustration purposes, however in practice their origins are both at the centre of the camera (C).

where q_α is additive measurement noise. The elevation (β) is the angle between the landmark point measurement and the body-fixed y-axis (or camera X_C-axis), as viewed from the perspective of the camera,

$$\beta = (\arccos(x_B/R) - \pi/2) + q_\beta, \tag{4.27}$$

where q_β is additive measurement noise. The measurements are presented in vector form and denoted by $\boldsymbol{z}_t$

$$\boldsymbol{z}_t = \begin{bmatrix} R_1 & \alpha_1 & \beta_1 & R_2 & \alpha_2 & \beta_2 & \dots & R_i & \alpha_i & \beta_i \end{bmatrix}, \tag{4.28}$$

where i is the number of landmark measurements in a single image.

4.3.1 Measurement Noise

Vision sensors are just as prone to noise as every other sensor used in this system. It is therefore necessary to develop a noise model to account for variations in measurements caused by sensor noise to ensure accurate system modelling. The vision sensor noise was determined by taking many image samples over a certain period of time where the multi-rotor system was completely stationary. The output of the measurements is the translation to the landmark reference point in meters, denoted by $X_C Y_C Z_C$ coordinates from the perspective of the camera, which was calculated by the *SolvePnP* function in the OpenCV image processing libraries. The translation vector in Cartesian coordinates was then converted to spherical coordinates with Equations 4.25-4.27, excluding the additive noise.

The spherical coordinate equations are non-linear, and therefore the measurement noise model will also be non-linear in nature. Therefore, several sets of samples were taken at various distances from a single landmark to develop an accurate noise model which can represent various changes in distance from the landmark.

The spherical coordinate system is used to represent landmark measurements since it provides the simplest and most independent description of noise in the measurements. The methods used in the *SolvePnP* function in OpenCV determines the pose of the camera relative to a landmark by measuring the distances between points in the image. The points are located on corners which are detected as features on the landmark, where the range to each point, as well as the noise in the range measurement, depends on the distance from the camera to the landmark. The azimuth and elevation are angles measured between pixels on the image plane, and are also dependant on the distance to the landmark; however, the noise in the angle measurements remains constant, since the noise on the image plane is constant. Therefore the assumption of independent noise is made owing to the constant error in angle measurements and the independent calculation of range noise. Since the range noise is not constant, it is necessary to find the relationship between the noise in the range measurement and the distance to the landmark. Consider the projection of two points on to the image plane of a camera (illustrated in Figure 4.6).

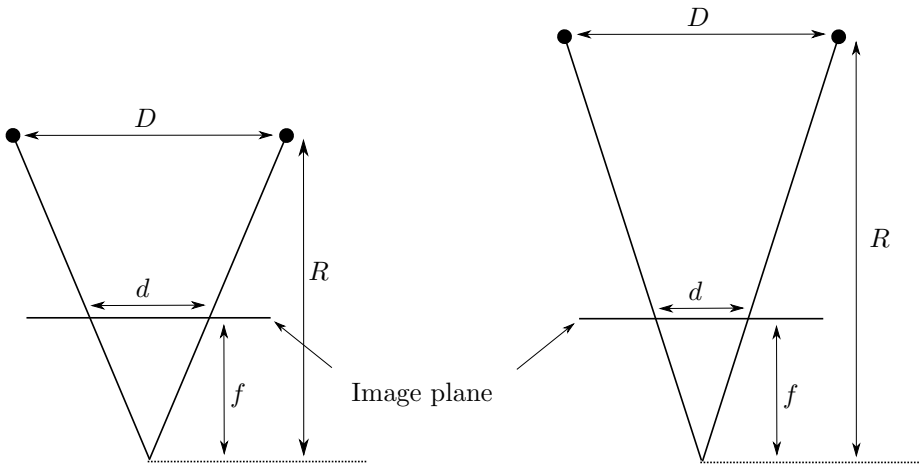


Figure 4.6: 2D two-point projection

If the camera moves further away from the points (R increases), the distance between the points (D) remains the same, however the distance between the projected points (d) on the image plane moves closer together, whilst the focal length (f) of the camera stays constant. Therefore, from basic trigonometry, it can be shown that

$$d/f = D/R, \tag{4.29}$$

and rearranging with respect to R

$$R = Df/d = k/d \qquad \text{where} \qquad k = Df, \tag{4.30}$$

where k is a constant. The relationship between R and d can therefore be plotted according to Figure 4.7. It is clear that the range measurement is a function of the distance being measured

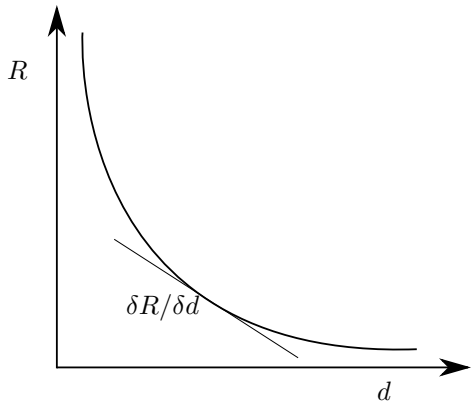


Figure 4.7: Relationship plot between range and pixel distance.

between the projected points on the image plane (which is measured in pixels). The relationship between R and d can be approximated using a Taylor series expansion which linearises the function at a certain point on the curve in Figure 4.7. This is done by calculating the derivative of the function in Equation 4.30 with respect to d,

$$\delta R/\delta d = -k/d^2. \tag{4.31}$$

Rearranging Equation 4.29 with respect to d, and substituting d and k it into Equation 4.31 yields

$$R' = -k/d^2 = -Df/(Df/R^2) = -R^2/Df. \tag{4.32}$$

The measurement noise in range is therefore not constant and needs to be calculated for each landmark measurement. If σ_R is the standard deviation in the noise for a measurement in range, then the noise variance can be calculated according to

$$\sigma_R^2 = (R^2/k)^2\sigma_d^2 = CR^4, \tag{4.33}$$

where C represents all the constants in the equation and R is the range specific to that measurement. The constant C was derived from several sets of sample measurements that were taken at various distances from landmarks to account for non-linearities in the range variance. The measurement noise can therefore be represented by the diagonal covariance matrix R_m

$$\mathbf{R_m} = \begin{bmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 \\ 0 & 0 & \sigma_\beta^2 \end{bmatrix}, \tag{4.34}$$

where σ_α^2 is the azimuth noise variance and σ_β^2 is the elevation noise variance which are both constant and independent of the range to the landmarks. If the measurement update step in the UKF algorithm is

$$\mathbf{y}_t = \boldsymbol{h}(\mathbf{x}_t, \boldsymbol{z}_t) + \mathbf{v}_t, \tag{4.35}$$

then the additive measurement noise is represented by $\mathbf{v}_t$, where it is drawn from a normal distribution calculated from the measurement noise covariance matrix

$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_m). \tag{4.36}$$

4.4 Summary

The concept of a kinematic motion model was introduced, followed by a description of the states. The mathematics behind Euler angle transformations were briefly explained, with specific discussions of the transformations that are performed on sensor data in this project. The kinematic motion model and image sensor measurement model was defined, which was the focal point of this chapter. The purpose of each model is to describe the relationship between the system states and the measurements provided by the various sensors. The motion model is related to the states of the multi-rotor through orientation and velocity sensor measurements, whereas the measurement model is related to the states through the image measurements. The mathematical description of these relationships enables the use of a filter, such as the UKF, to perform state estimation on the system, provided that process and measurement noise models are well defined.

Chapter 5

Image Processing

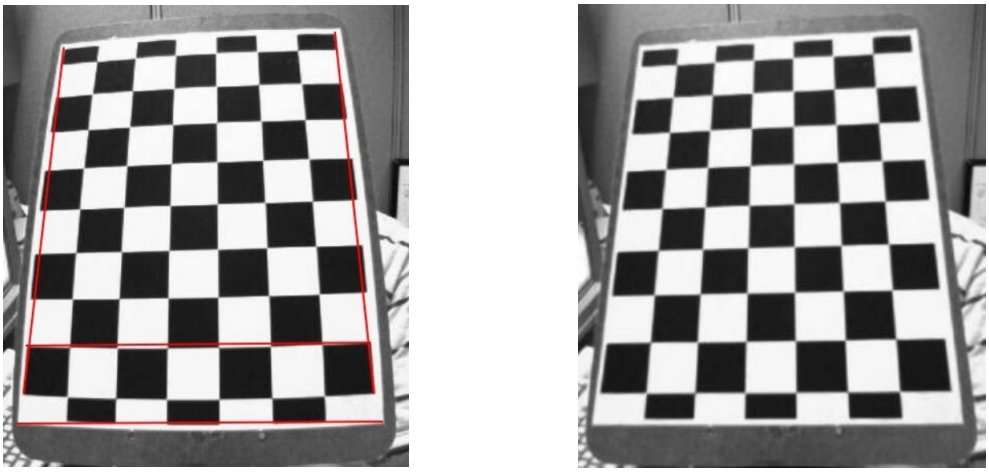
The most important sensor on the multi-rotor system is the single camera, which takes measurements in the form of images. These images consist of thousands of pixels, each of which have a specific colour intensity value and image coordinates. The entire realm of image processing relies on mathematical calculations performed on a pixel (or groups of pixels) which are identified to fall within a certain range of colour intensity. To simplify the grouping of pixels with a certain colour intensity, the pixels are converted from the red-blue-green colour spectrum to greyscale. In greyscale, each pixel has a colour value that ranges between zero and 255, the number being representative of the shade of grey that occupies the pixel, from white all the way to black. The greater the difference in greyscale value between one pixel and the neighbouring pixel, the easier it is to determine point-features in the image. These point features have specific image coordinates which are measured in number of pixels along the $x_c y_c$ image axes (previously illustrated in Figure 2.4). Once the image coordinates of point features (which may form part of a landmark) are identified, the 3D coordinates of those points can be calculated through the methods derived from the pinhole camera model. Therefore, the position of the camera relative to the point features on the landmark can be calculated. Furthermore, the way artificial landmarks are designed determines how effectively point features can be detected by image processing algorithms. Landmark design is therefore also an important consideration where the accuracy of a localisation algorithm is dependent on the reliability and effectiveness of feature detection on artificial landmarks.

5.1 Camera Calibration

Before any type of image processing can be performed, it is necessary to calibrate the camera in order to determine its intrinsic parameters. The intrinsic parameters are unique to each camera and are also constant, it includes information like the camera matrix and distortion coefficients. The intrinsic camera parameters are determined by taking several pictures of the same checkerboard landmark from various angles and putting those images through OpenCV's camera calibration function. This function uses the principles of the pinhole camera model in §2.4 to determine the value of each intrinsic parameter. The camera matrix contains the focal lengths and optical centres of the camera, whereas the distortion coefficient matrix contains the variables that are used to undistort images to compensate for the fish-eye effect some lenses may have. The optical centre of the camera, which is assumed be in the centre of the image plane, is considered as the point around which distortion from the camera lens occurs, which is referred to as radial distortion. Additionally, there may be a

misalignment between the image sensor and the camera lens which causes another type of distortion, referred to as tangential distortion [24].

The checkerboard landmark that is used for calibration has exact known parameters, such as the number of squares in each row and column, as well as the size of each square. This information can be used to compare the perfect pinhole camera model projection of the landmark (with no lens distortion) to the actual projection which is somewhat distorted by the curvature of the camera lens. The error between the two projections is used to calculate the parameters in the camera and distortion matrices, which can then be used to undistort images that were taken by that camera, as illustrated in Figure 5.1.



(a) Image with radial distortion

(b) Image after being undistorted

Figure 5.1: (a): An image of a checkerboard landmark where radial distortion is present. (b): The result of undistorting the image by making use of the intrinsic camera parameters which were obtained through the camera calibration process [49] (Reproduced under the Creative Commons License).

Once the camera is fully characterised, image processing can be performed on any of the undistorted images. The 3D pose of the camera can then be calculated from projected points on the image plane by making use of principles derived from the pinhole camera model, which are only applicable to the undistorted images. Therefore, if the camera is calibrated correctly, the resulting information can be used to develop an accurate mathematical model of the camera. The accuracy of the results produced by OpenCV's camera calibration function may vary depending on the type of checkerboard pattern that is used. A larger square size will generally result in a more accurate camera calibration, for this reason a 9x7 pattern was used with 10 centimetre square sizes. Furthermore, in order to get the most accurate results, the camera calibration process was repeated for several data sets of images, taken at varying distances and orientations, the average of which was used as the intrinsic parameters.

5.2 Landmark Design

Any point feature that is identified by image processing can be considered as a landmark; however, consistently detecting the same point feature in several consecutive images is a very challenging task. To simplify this task, a feature rich landmark is required, with particular features that make it easily detectable in every image frame. Additionally, the landmarks should be large enough to be detectable from the maximum expected distance from the camera (which is roughly three meters). Another important requirement in this localisation problem is the ability to repeatedly identify multiple unique features or landmarks. These requirements warrant the design of multiple unique and robust landmarks which can be used in this thesis.

One of the simplest and most effective features used in image processing is a corner where two highly contrasted groups of pixels meet, such as black and white squares. The two highly contrasted colours reduce the possible colour intensity value in each pixel from a wide range of 255 values to an easily detectable binary difference, which improves the robustness of an image processing algorithm searching for landmarks. A common landmark pattern which uses black and white squares is the checkerboard (herafter referred to as a chessboard in accordance with OpenCV's terminology). This type of pattern is used extensively in landmark designs as well as camera calibration for computer vision applications.

For the localisation application in this thesis, several unique landmarks are required; therefore, from the literature review in §2.1, there are few viable options for the type of landmark that can be used. The fixed ball-pattern landmark would be cumbersome to manufacture, especially in multiple unique variations, and the concentric squares and circles are not visually unique in orientation. The chessboard, however, is a versatile landmark type that can be applied in various scenarios with many visual features, it is also a highly supported landmark format in OpenCV image processing functions. For these reasons, a modified version of the chessboard landmark is used with an easily recognisable four bit binary number which makes each landmark unique. The binary number needs to be represented in a way that would make it easily identifiable from various distances through image processing techniques; therefore, each number is represented by a pattern of black and white circles. Each circle represents a binary digit, with the most significant bit on the right, as illustrated in Figure 5.2a, where a white circle represents a zero, and a black circle represents a one. Circles were chosen to represent the binary numbers to remove ambiguity which may be introduced by more corners, since a landmark's orientation or point of interest may be incorrectly identified by inconsistent corner features. Equation 5.1 shows an example of how a landmark's identity is calculated if it has the following circle pattern: black, white, black and black,

$$BWBB : 1011 = 2^0 + 0^1 + 2^2 + 2^3 = 1 + 0 + 4 + 8 = 13. \tag{5.1}$$

The four-bit binary number pattern used to identify landmarks can represent a maximum of 16 unique landmarks, which are placed in a particular pattern. Owing to the altitude restriction of the flight path and the limited number of unique landmarks, the landmark pattern can only be placed over a limited area whilst keeping all of the landmarks in the field of view of the camera. However, the number of unique landmarks that can be represented can be increased to 32 if a five-bit binary number is used for landmark identification. In the case where 16 unique landmarks are insufficient, 32 unique landmarks can be trivially realised by using five black or white circles in place of four.

The size of each square is precisely seven centimetres, which fits on to an A3 size landmark with five rows and four columns of squares. The 5x4 square pattern provides 12 corner features

on each landmark, this allows for a single solution to the perspective-n-point problem (discussed in §2.5) which describes the pose of the camera relative to the landmark in six degrees of freedom. Furthermore, this exact square size was chosen to maximise the size of the projected pattern on the image plane, which results in higher resolution image measurements. Since the noise in the image plane remains constant, the pixel distances between projected points are represented by more pixels, resulting in a higher resolution measurement with the same amount of sensor noise. An important constraint with regards to landmark placement is that of the camera's field of view, the Raspicam used in the data acquisition has a very narrow field of view (FOV). The narrow FOV may result in only partial landmark visibility if the landmarks are too large, and a landmark will only be recognised if the entire landmark falls within the FOV of the camera. If a landmark size larger than A3 was used, there would be more instances where landmarks are partially visible in the image frame owing to the altitude constraint of the multi-rotor's flight path.

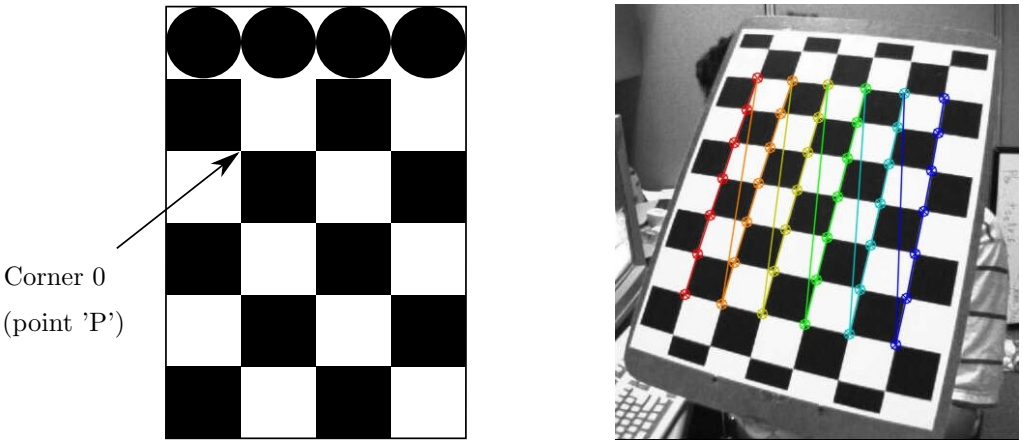
5.3 Landmark Detection

Implementing image processing methods and techniques from scratch is an extremely complex task and in most cases not necessary, since there are an abundance of open source image processing libraries available for academic use. In this thesis the OpenCV Python libraries are used to implement image processing techniques to detect and identify unique landmarks in the environment. OpenCV in Python is perfectly suited as a prototyping tool for image processing implementations and contains several built-in features, such as camera calibration and chessboard detection functions. These functions make use of a plethora of methods and techniques to detect various features in images which may be of use in further calculations.

One of the most used features in image processing is a corner, which is an easily identifiable feature where two or more edges meet. One of the fundamental building blocks of image processing is the concept of a gradient in the image, which describes the direction of the change in colour intensity from one pixel to the next. If the greyscale intensity of one pixel is higher than its neighbouring pixel, then the gradient moves in the direction of the darker pixel. The concept of gradients can therefore be used to describe the area at which two edges meet (such as a corner), which is usually subject to some change in colour intensity. The change in colour intensity can be identified by the high variation in the gradient of the image in that area, which can be used to detect corner features. Each corner feature that is detected in an image has a position in the image plane that can be described by pixel coordinates. The accuracy of the corner feature's pixel coordinates may vary, depending on how high the variation in gradient is between the edges. If the change in colour intensity is low, there is a lot of ambiguity in the exact position of the corner, which may be described by an area of many pixels. However, if there is a very high contrast in the colour intensity, the position of the corner is represented by much fewer pixels, and therefore at a much higher accuracy. The highest possible variation in gradient, and therefore most accurately defined corners, is represented by black and white squares.

OpenCV has a built in function which is designed to find chessboard corners in an image (called *findChessboardCorners*), given the number of squares in each row and column. This function determines the pixel coordinates of each corner in a very specific order (illustrated in Figure 5.2b). The function operates under certain assumptions: that all the corners are on a flat surface in the same plane, and that all squares are rectangular and equidistant from one another. This knowledge is later used to calculate the position and orientation of the camera with respect to the landmark by

solving the perspective-n-point problem. The pixel coordinates of each corner is stored in a matrix, where the first entry in the matrix will always be the same corner, it can therefore be used as a consistent point of reference over multiple images of the same landmark. In this thesis the point of reference on each landmark is corner 0, as indicated in Figure 5.2a. The *findChessboardCorners* function automatically detects a single visible chessboard pattern in the image, which leaves any other chessboards undetected. Therefore, the function needs to be executed multiple times for a single image frame to detect all of the visible landmarks. To avoid double detection of the same landmark, a neutral coloured polygon is drawn over a landmark after it has been detected (removing the landmark from the image). This enables multiple detections of unique landmarks in a single image and avoids double detection of the same landmark, which is computationally expensive.



(a) Landmark number 15

(b) OpenCV chessboard detection [49]

Figure 5.2: (a): An example of the exact landmarks designed for this thesis project, where corner 0 is the reference point for that landmark (point 'P' as described in Figure 2.4), and the binary number identifies it as landmark 15 (as per Equation 5.1). (b): The result of OpenCV's chessboard corner detection function, visually illustrating the order in which corners are detected, starting at the red corner in the bottom left of the chessboard.

OpenCV has another important function called *solvePnP*, which is designed to calculate the camera's pose by point correspondences in the image plane, given certain parameters and assumptions relating to the points. The *solvePnP* function uses mathematical principles based on the pinhole camera model (previously discussed in §2.4 and §2.5) to determine the extrinsic parameters of the camera, which describes the rotation and translation of the camera relative an object on which the correspondent points are found. Therefore, the assumption when using *solvePnP* in combination with *findChessboardCorners*, is that each corner that is detected on the object has known positions relative to every other corner. This is the case for the chessboard landmarks that are used in this thesis, owing to the assumptions under which *findChessboardCorners* operates. Since each landmark contains 12 corners, there are enough point correspondences to solve for the pose of the camera in six degrees of freedom in this specific perspective-12-point problem. Each corner on the landmark, which is initially described by pixels in the image plane, can then be described as a 3D point by the translation matrix that results from the *solvePnP* function. The translation matrix describes the Cartesian coordinates of each corner as a point in the camera axes (X_C, Y_C and Z_C). Therefore, any

single corner can be chosen as a point measurement to that specific landmark (in this thesis corner 0 is used). Once the Cartesian translation vector to the landmark point reference is known, a 3D measurement to that landmark can be calculated in spherical coordinates in the camera reference frame. The spherical 3D coordinates are calculated by making use of Equations 4.25 - 4.27 in the measurement model, the same equations that are used to predict the landmark measurements from the multi-rotor states in §4.3 (however no additive noise is included, since the actual measurement will already contain sensor noise).

Once a landmark has been detected in the image, the unique identity of that landmark can be determined by decoding the binary information in the black and white circles. An additional landmark identification function was coded for this purpose, since there is no built-in OpenCV function that could perform the task. The function first samples the greyscale value of the black and white squares closest to the circles to determine a base-colour, since the intensity of a black greyscale value may vary, depending on the light conditions. The image coordinates of each corner on the chessboard is known, since it was calculated by the *findChessboardCorners* function; furthermore, the position of all the circles relative to the first row of squares are also known from the landmark design. Therefore, by making use of the knowledge of each corner's image coordinates, it is possible to determine the image coordinates of the centre of each circle by means of extrapolation, since the pixel size of the squares and circles are known. Thereafter, the landmark identification function samples the greyscale value in the centre of each circle and compares the intensity to the base-colour. If the greyscale value of a circle's sample area falls within a certain range of the base-colour, it is classified as either a one or a zero, and thereby the binary identity of the landmark is determined. This process of landmark detection and classification is repeated for each landmark that is fully visible inside the image frame, the result of which is shown in Figure 5.3. The process of landmark detection and identification to generate sensor measurements can be illustrated by the following pseudo code:

```
> Store corners <- find chessboard in image
> while chessboards are found in image:
>    Store translation to chessboard <- solve PnP function
>    Store landmark vector <- convert translation from cartesian to spherical
>    Store identity of landmark <- identify the landmark
>    Remove landmark from image
>    Search for another landmark in the image
>    If found, repeat loop
>    ElseIf no landmark found, exit loop
```

The landmark detection algorithm will search for a landmark by finding corner features that are arranged in a specific pattern; if a landmark is found, it will find the translation matrix of the camera relative to each corner on the landmark. Thereafter it will convert the X_C, Y_C and Z_C Cartesian coordinates of corner 0 on the landmark to spherical coordinates in range, azimuth and elevation. Once the landmark has been identified, the landmark measurement is added into a matrix holding all of the measurements (one for each landmark that is visible in the image), after which the landmark is removed from the image by drawing a polygon over it. The algorithm then searches for another landmark, and the process is repeated for each visible landmark until no more landmarks are detected.

All 15 of the landmarks are accurately placed in a staggered pattern at known positions in the

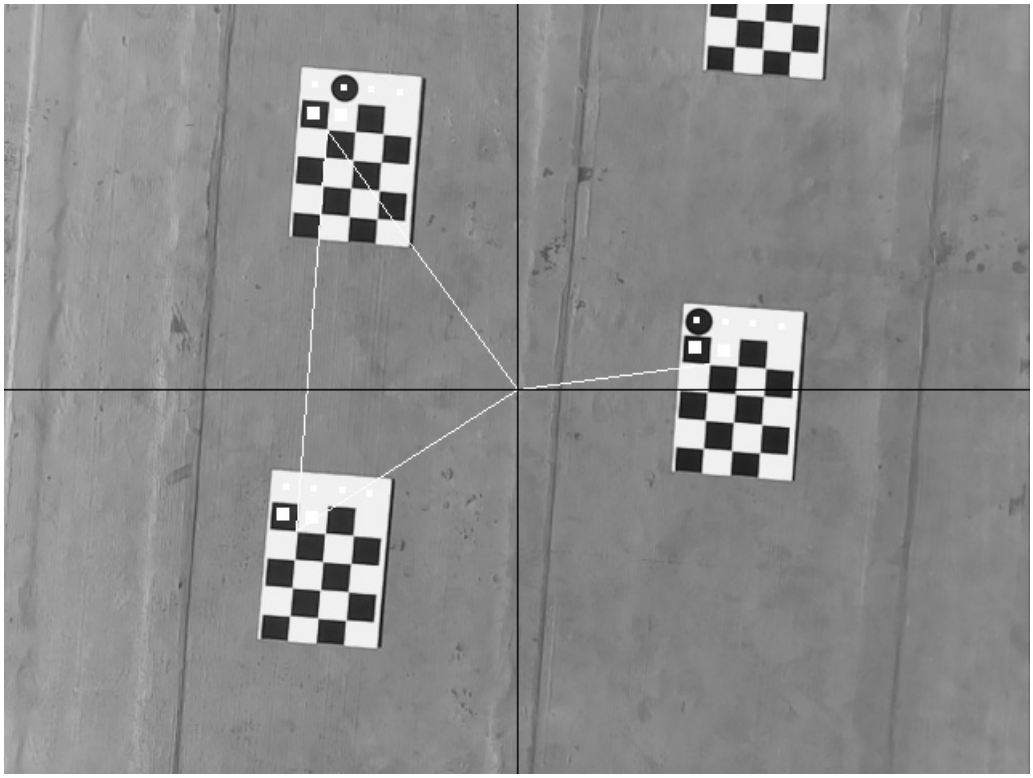


Figure 5.3: An example of multiple landmarks that are detected in a single image, where the landmarks are identified as zero, one and two. The small white squares inside the black circles indicate the area where the colour of that bit is sampled to determine the landmark identity. The lines protruding from the centre of the image axis visually illustrate the relative position of each landmark's reference corner.

---

environment, where corner zero on landmark zero is the origin of the landmark axes. All of the even numbered landmarks therefore lie exactly on the Y-axis, this information is used to determine the initial landmark alignment angle (discussed in §6.3). The lines protruding from the centre of the image in Figure 5.3 is the measured relative displacement of each landmark reprojected on to the image plane, and it also indicates that the landmarks have been detected correctly. Each landmark's 3D position is converted from Cartesian coordinates to spherical coordinates in the camera reference frame, and is stored in a matrix along with the landmark's identity, which is later used in the UKF state estimator in §6.2.

5.4 Measurement Noise Verification

Another aspect of accurate landmark detection that requires consideration is the influence of noisy measurements, this is overcome by modelling the measurement noise (as in §4.3.1). However, the characteristics of the measurement noise may not be the same in practice as in theory. To model the measurement noise and verify that the practical findings hold true to the theoretical expectations, multiple data sets of measurements were taken of the same landmark, each at an increased distance from the landmark. The camera was completely stationary during the acquisition of each data set,

therefore any variation in the landmark measurements can be attributed to image sensor noise. The measurements were processed and converted to spherical coordinates in the landmark reference frame by the same algorithm described in §5.3. The standard deviation in the spherical measurements can be used to calculate the noise variance; however, the noise variance in range is expected to scale with a constant C that is multiplied with the fourth power of the magnitude of the range measurement, as derived in Equation 4.33 (repeated here for clarity),

$$\sigma_R^2 = CR^4. \tag{5.2}$$

The scaling constant C was solved for by taking the measured noise variance as well as the average of the measured range, and rearranging Equation 5.2 with respect to C. If the equation for noise variance in range is a realistic approximation, the value of C is expected to be similar for each data set. The values for C at one and three metre distances from the landmark was found to vary by only 10%; therefore, the average of C at all sample distances were taken as the final constant value, where

$$C = 1.5e^{-8}. \tag{5.3}$$

The noise variances in azimuth and elevation angles was expected to stay constant, owing to the constant noise in the image plane. The expected behaviour was confirmed from the calculated noise variances in each respective data set, where the noise variance at each distance varied by roughly 10% as well. Figure 5.4 shows the range noise variance calculated for each range value, plotted against the actual measured range noise variance, both of which show an increase in noise variance with an increase in distance from the landmark. The deviation of the measured noise from the calculated noise in Figure 5.4 can be attributed to the approximation of the constant C which is used to scale the noise variance with range. However, the maximum difference between the calculated and measured noise variance at a range of three metres results in a standard deviation difference of only 0.3 centimetres. It is clear that this noise model is only a suitable approximation for a certain range of distances from the landmark, below three metres. This noise model is therefore sufficient for the application in this thesis where the multi-rotor is limited to an altitude below three metres.

5.5 Limitations

Image processing is subject to various limitations and constraints, and it is especially important to consider these in applications where the robustness of the image processing algorithm has a direct impact on the success of a localisation problem. If a landmark is not detected it could result in less accurate state estimation; however, if a landmark is detected incorrectly it could result in a completely incorrect estimation of the states. Therefore, it is necessary to understand the limitations of the image processing algorithm in order to implement it to be robust and effective.

The most important sensor is the on-board camera, and the outcome of its image measurements rely significantly on the surrounding light conditions. Poor light conditions may have a negative effect on landmark detection if glare is present in the image of the landmark, or if low light makes the black and white squares appear less distinguishable from one another. To avoid image deterioration owing to poor light conditions, flight tests were conducted on sunny days in an open area, and landmarks with matt-vinyl prints were used to avoid glare. Several other factors could also cause deterioration of landmarks in images, such as high altitude and instability of the camera; however these effects

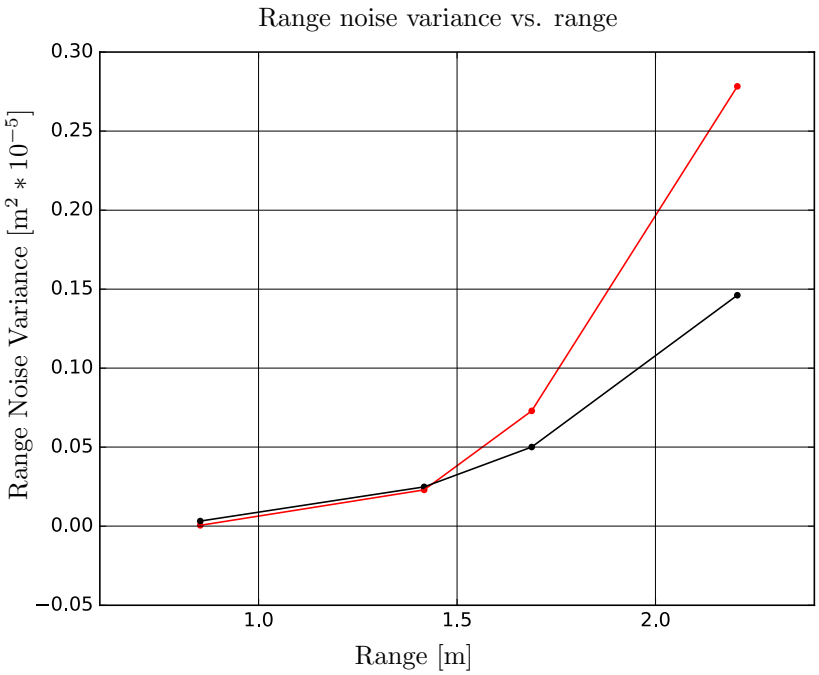


Figure 5.4: The calculated range noise variance (black) plotted against the measured range noise variance (red). Each sample set was taken at an increased distance from the landmark, between one and three meters away. The calculated value follows the trend of the measured value, however at further distances from the landmark, the measured noise variance deviates from the calculated noise variance.

---

did not play a role owing to the low altitude flight path requirement for antenna characterisation. Overall, few other limitations were encountered with regards to image processing results on the data sets, and incorrect detection of landmarks happened on rare occasions that can be contributed only to non-repeatable phenomena.

5.6 Summary

The application of camera imaging sensors in a localisation problem was introduced, with specific reference to the requirement of visual features that can be used as landmarks. The process of the landmark design for this project was discussed, which motivates the black and white chessboard pattern. The image processing methods that enable detection of the chessboard landmarks were explained, describing how each chessboard is used as a single point-reference landmark to generate a measurement. Finally, the measurement noise model designed in the previous chapter is verified, followed by a brief discussion of the limitations that image measurements are typically subject to.

Chapter 6

UKF State Estimation

State estimation is one of the most prolific topics in academic studies on robotic systems. No autonomous system can function without an accurate description of its current states, whether the environment is known or not. Some of the most commonly used methods in state estimation comes from the family of Kalman filters. In this thesis, a variant of the traditional Kalman filter is used, called the unscented Kalman filter. This section explores the inner workings of the UKF and how it brings together all of the various subsystems to perform state estimation.

6.1 Overview of State Estimation

Each state of the multi-rotor can be represented by a random variable, which always has a specific, finite value that is never exactly known but can be estimated. The knowledge of a state is given by a probability distribution of all the possible values of the random variable. The probability distribution (which is also called the belief distribution) therefore represents a range of random variable values which could hold true for that state, where the distribution is Gaussian in nature. All of the material and equations discussed in this chapter is based on work by Thrun [26].

6.1.1 Gaussian Distributions

The Gaussian probability distribution can be graphically represented by a probability density function (PDF, illustrated in Figure 6.1), and is characterised by two parameters, namely the mean and variance. The mean of the Gaussian distribution is the expected value of the random variable that is characterised by the probability distribution. Therefore, the mean (μ) of the Gaussian belief distribution of a state represents the random variable value that has the highest likelihood of being a true reflection of reality. The variance of the Gaussian belief distribution (σ^2), which is the standard deviation (σ) squared, indicates how far the belief distribution is spread from the mean (as shown by the dotted lines in Figure 6.1). If the standard deviation (and therefore the variance) is larger, the Gaussian distribution would be wide and flat, which is indicative of a very uncertain state estimate. However, if the variance is smaller, the Gaussian distribution would be taller and narrower, which is indicative of a much more certain state estimate. The distribution plotted in Figure 6.1 can be described by Equation 6.1

$$\mathcal{N}(\mu, \sigma) = (\sqrt{2\pi\sigma^2})^{-1} \mathrm{e}^{-(x-\mu)^2/2\sigma^2}. \tag{6.1}$$

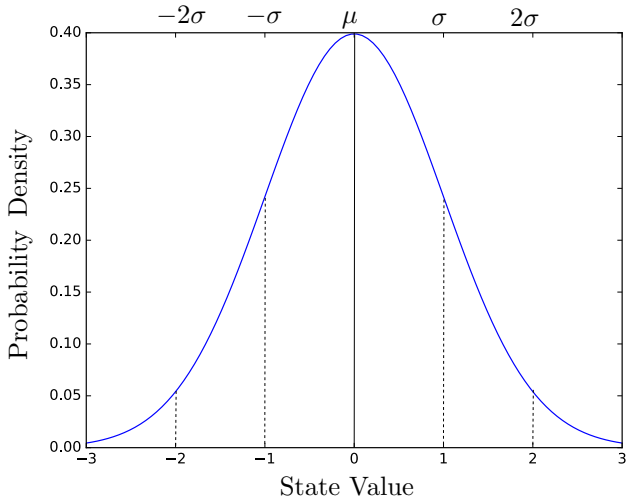


Figure 6.1: An illustration of a one dimensional Gaussian belief distribution of a certain state, also called a probability density function (PDF). The PDF in this figure has a mean of 0 and a variance of 1, and the dotted lines represent the one- and two-sigma confidence ranges.

The Gaussian distribution (also called the normal distribution) describes the probability of a scalar random variable having a certain value. However, states are often described by vectors which represent values in more than one dimension, in which case the univariate Gaussian distribution is insufficient. PDFs can also be used to describe vectors of random variables in multiple dimensions, however their graphical representation is different, as illustrated in Figure 6.2 for two dimensions. The Gaussian distribution over a vector is called multivariate, where the distribution in Figure 6.2

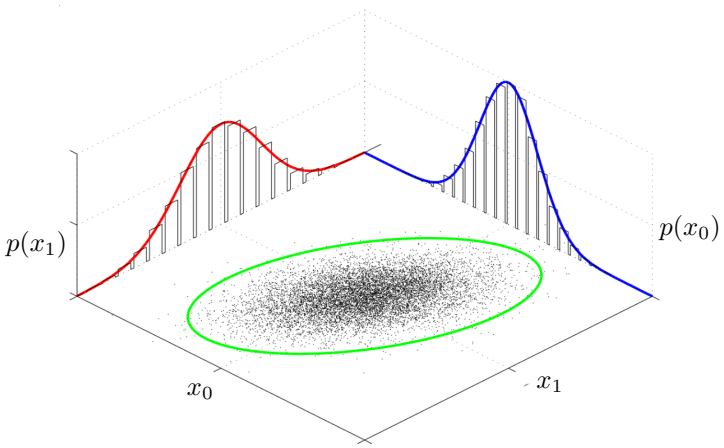


Figure 6.2: A Multivariate Gaussian distribution in two dimensions, where $p(x)$ denotes the PDF for the relevant random variable. The ellipse between the two Gaussian plots is a common graphical representation for multivariate distributions, where the shape of the ellipse gives some indication of the covariance matrix related to the random variables (reproduced under the Creative Commons License [50]).

---

can be characterised by a different function [26],

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \det(2\pi\boldsymbol{\Sigma})^{-1/2} \exp\left((-1/2)(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{6.2}$$

Where $\boldsymbol{\mu}$ is the mean, and $\boldsymbol{\Sigma}$ is the covariance matrix (analogue to the variance, but for multiple dimensions of data). The random variable that represents the vector can be written as

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{6.3}$$

The covariance matrix ($\boldsymbol{\Sigma}$) describes the extent to which multiple random variables will change together, and it is used to calculate the correlation between the random variables that represent the states. The variation in a collection of random points in several dimensions (such as the Gaussian belief distribution of two random variables in Figure 6.2) cannot be characterised by a single number. Therefore, the multi-dimensional variance is described by the symmetrical, n-dimensional covariance matrix $\boldsymbol{\Sigma}$, where n is equal to the length of the state vector $\mathbf{x}_t$. The covariance matrix holds information about the variances of each state and their relationships with one another, where the variance of each state variable falls on the diagonal in the matrix.

6.1.2 State Estimation Filters

The purpose of filters in state estimation is to implement methods and techniques to best approximate the value of a random variable, given information on the previous (also called the posterior) states and current sensor values. This is done by propagating multivariate belief distributions of the state vector through certain functions (which are often non-linear), given certain sensor parameters of the system. The Bayesian family of filters are some of the most widely used filters in state estimation, where the Bayes filter algorithm is a general probabilistic approach to calculating probability density functions given specific information about the states and measurements of the system. The Bayes filter defines some of the principles upon which many complex algorithms in its family are built. An important consideration in the Bayesian filter family is the Markov assumption (also called the complete state assumption). The Markov assumption says that the previous state and the next state is independent, if the current state $\mathbf{x}_t$ is known. This assumption allows for simplification in the representation of states to reduce the computational complexity of filters in the Bayesian family.

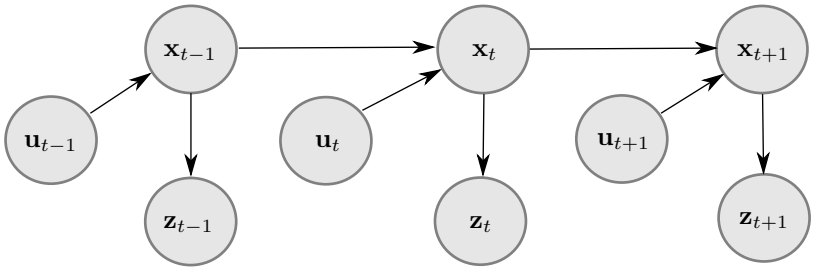


Figure 6.3: The dynamic Bayes network which characterises the evolution of controls, states and measurements. The Markov assumption is illustrated here, where it is clear that if you know $\mathbf{x}_{t-1}$, then $\mathbf{x}_{1:t-2}$ does not add any new information about the current state. Another assumption that is made in the Bayes network is that the measurement $\boldsymbol{z}_t$ is conditionally independent of all previous states, measurements and control inputs, given $\mathbf{x}_t$.

The Bayes filter is the most general algorithm which can be used to calculate belief distributions over states, whereby it calculates the belief distribution (denoted by bel) by making use of control and measurement data. The belief over a state $\mathbf{x}_t$ is denoted by $bel(\mathbf{x}_t)$, which is a function of the current state, given all the measurements and control inputs from $t = 1$,

$$bel(\mathbf{x}_t) = p(\mathbf{x}_t|\boldsymbol{z}_{1:t}, \mathbf{u}_{1:t}), \tag{6.4}$$

where $\boldsymbol{z}_{1:t}$ refers to the sensor measurements and $\mathbf{u}_{1:t}$ the refers to the control inputs. Furthermore, the predicted belief (which is calculated before the newest measurements are incorporated) can be denoted by

$$\overline{bel}(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{u}_{1:t}, \boldsymbol{z}_{1:t-1}), \tag{6.5}$$

where $\boldsymbol{z}_{t-1}$ are the sensor measurements up to the previous time step. The Bayes filter is recursive, which means that its current belief over a certain random variable $bel(\mathbf{x}_t)$ at time t is calculated from the belief $bel(\mathbf{x}_{t-1})$ at the previous time step $t-1$. Owing to the Markov assumption, the historical data for measurements and control inputs are not considered, and only the posterior and current time step's information is used. The Bayes filter's input is therefore the belief over the state at the previous time step $t-1$, the current control input $\mathbf{u}_t$ and the current measurement $\boldsymbol{z}_t$. The recursive nature of the Bayes filter and the propagation of information is shown in Figure 6.3, where the system states are represented by $\mathbf{x}_t$. The belief over a random variable $bel(\mathbf{x}_t)$ is determined by first calculating the predicted belief $\overline{bel}(\mathbf{x}_t)$ over that state by using the previous state belief distribution $bel(\mathbf{x}_{t-1})$ and the current control input

$$\overline{bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \tag{6.6}$$

The motion model $\boldsymbol{g}$ that defines the propagation of states (discussed in §4.2) is described by the conditional PDF

$$p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}). \tag{6.7}$$

The function $\boldsymbol{g}$ therefore calculates the current state $\mathbf{x}_t$ that has the highest probability of being true, given that the kinematic sensor data $\mathbf{u}_t$, and previous states $\mathbf{x}_{t-1}$ are true. The predicted belief distribution that is assigned to the current state is obtained by the integral of the product of the two distributions. This step is referred to as the *control update* or *prediction*, since it calculates the belief over a state based on the control input and the previous state's knowledge. Thereafter, the current belief $bel(\mathbf{x}_t)$ is calculated from the predicted belief distribution of the state, as well as the measurements resulting from the current state

$$bel(\mathbf{x}_t) = \eta p(\boldsymbol{z}_t|\mathbf{x}_t) \overline{bel}(\mathbf{x}_t). \tag{6.8}$$

This step is called the *measurement update*, since it multiplies the predicted belief $\overline{bel}(\mathbf{x}_t)$ by the likelihood that the measurements $\boldsymbol{z}_t$ may have been given. These calculations are done for every hypothetical posterior state, and the result is normalised by the normalisation constant η which leads to the current belief $bel(\mathbf{x}_t)$. The measurement model $\boldsymbol{h}$ (discussed in §4.3) is described by the conditional PDF

$$p(\boldsymbol{z}_t|\mathbf{x}_t), \tag{6.9}$$

which calculates the expected measurements $\boldsymbol{z}_t$, given that the current state $\mathbf{x}_t$ is true.

6.2 The Unscented Kalman Filter

The UKF is a specific case of the Bayes filter algorithm, which takes the unscented transform approach to linearisation in order to describe state estimates in Gaussian form. This section discusses the concept of the unscented transform and how it fits into the UKF algorithm to perform state estimation.

6.2.1 Overview of the UKF

The UKF is similar to the Bayes filter in that it characterises the evolution of controls, states and measurements according to the dynamic Bayes network in Figure 6.3. The Markov assumption also applies to the UKF, where the current state belief is conditionally independent of all earlier data, given the immediately posterior state belief. Additionally, the current measurements are only dependent upon the current state, and are therefore conditionally independent of all previous states. Both the Bayes and UKF filters can only process Gaussian posterior distributions, where the UKF uses a technique called the unscented transform to estimate Gaussian distributions for non-linear motion and measurement models.

To get a better understanding of the UKF as a whole, it is necessary to understand how the UKF propagates information, which is summarised by the flow diagram in Figure 6.4. The estimated states

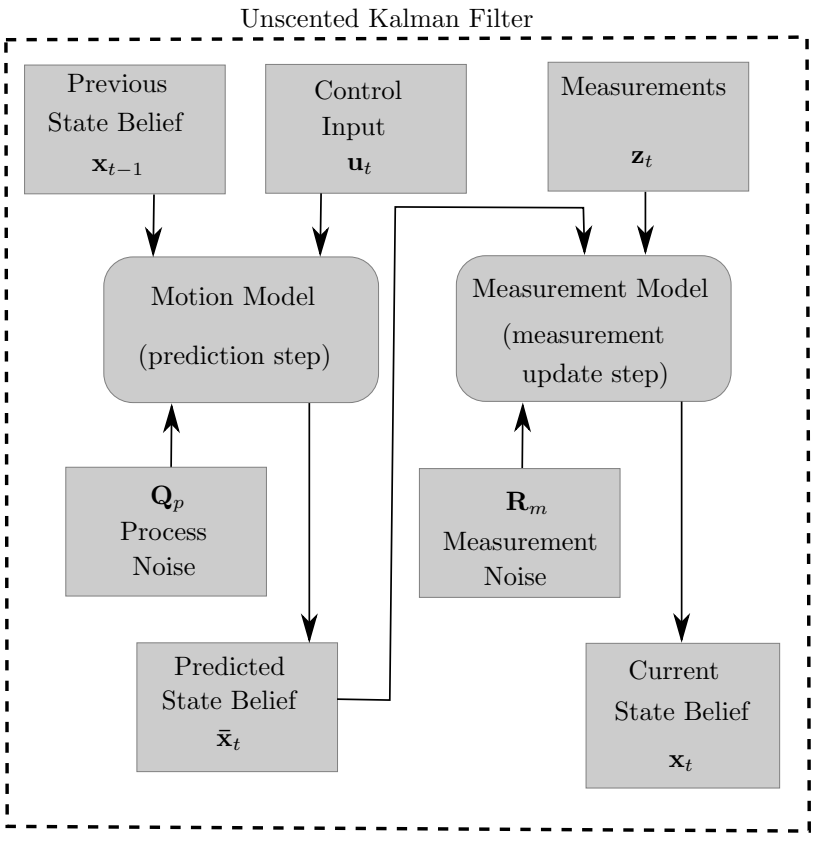


Figure 6.4: An overview of the UKF, showing the flow of information to determine the current state estimates.

of the multi-rotor at the previous time step $\mathbf{x}_{t-1}$ and the current control input $\mathbf{u}_t$ is propagated through the motion model $\boldsymbol{g}$ to predict the states at the current time step (prediction step)

$$\bar{\mathbf{x}}_t = \boldsymbol{g}(\mathbf{u}_t, \mathbf{x}_{t-1}) + \mathbf{w}_t. \tag{6.10}$$

Normally distributed additive noise $\mathbf{w}_t$ is introduced to the predicted current states. The current states are then propagated through the measurement model $\boldsymbol{h}$, which calculates the expected measurements $\hat{\boldsymbol{z}}_t$ that would result from the current states if they were an accurate reflection of the real pose of the multi-rotor (measurement update step)

$$\hat{\boldsymbol{z}}_t = \boldsymbol{h}(\mathbf{x}_t) + \mathbf{v}_t. \tag{6.11}$$

Normally distributed noise $\mathbf{v}_t$ is added to the expected measurements as well. The actual measurements of landmark positions $\boldsymbol{z}_t$ are determined by the image processing algorithm and are compared to the predicted measurements. The difference between the actual measurements and the predicted measurements is multiplied with a variable called the Kalman gain $\mathbf{K}_t$ to adjust the predicted states,

$$\mathbf{x}_t = \bar{\mathbf{x}}_t + \mathbf{K}_t(\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t). \tag{6.12}$$

The Kalman gain is a relative weight given to the predicted state estimates and the measurements, and is used to calculate the current state estimates $\mathbf{x}_t$. The final state estimates are the result of many intermediate calculations, such as that of the covariance matrix $\bar{\boldsymbol{\Sigma}}$, cross-covariance matrix $\bar{\boldsymbol{\Sigma}}_t^{x,z}$ and measurement uncertainty $\boldsymbol{S}$. However, the most important process is that of the unscented transform, which is an inherent characteristic to the UKF. The normal Kalman filter can only accommodate linear state propagation functions, therefore variations of the traditional Kalman filter exists which makes use of various linearisation techniques. The unscented transform is one of the linearisation techniques used to describe probability distributions in Gaussian form after they were propagated through non-linear functions. The unscented transform deterministically extracts a set of so-called sigma points from the Gaussian probability distribution of each state, thereafter the sigma points are passed through a certain non-linear function and used to calculate the new mean and covariance after the state propagation, as visualised in Figure 6.5. The number of sigma points that are selected for each state is $2n + 1$, where n is the number of states in the system.

The selected sigma points are symmetrically located around the mean and are chosen according to the following rule, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix:

$$\boldsymbol{\mathcal{X}}^{[0]} = \boldsymbol{\mu}, \tag{6.13}$$

$$\boldsymbol{\mathcal{X}}^{[i]} = \boldsymbol{\mu} + (\gamma\sqrt{\boldsymbol{\Sigma}})_i \quad \text{for } i = 1, ..., n \qquad \text{and} \tag{6.14}$$

$$\boldsymbol{\mathcal{X}}^{[i]} = \boldsymbol{\mu} - (\gamma\sqrt{\boldsymbol{\Sigma}})_{i-n} \quad \text{for } i = n+1, ..., 2n. \tag{6.15}$$

Here γ is a constant calculated by certain scaling parameters that determine how far the selected sigma points are spread from the mean of the probability distribution. The i denotes the column in the matrix that is being used to calculate the sigma point vector. In Equations 6.14 and 6.15 it is necessary to calculate the square root of the covariance matrix $\sqrt{\boldsymbol{\Sigma}}$, where the square root of a matrix is defined such that

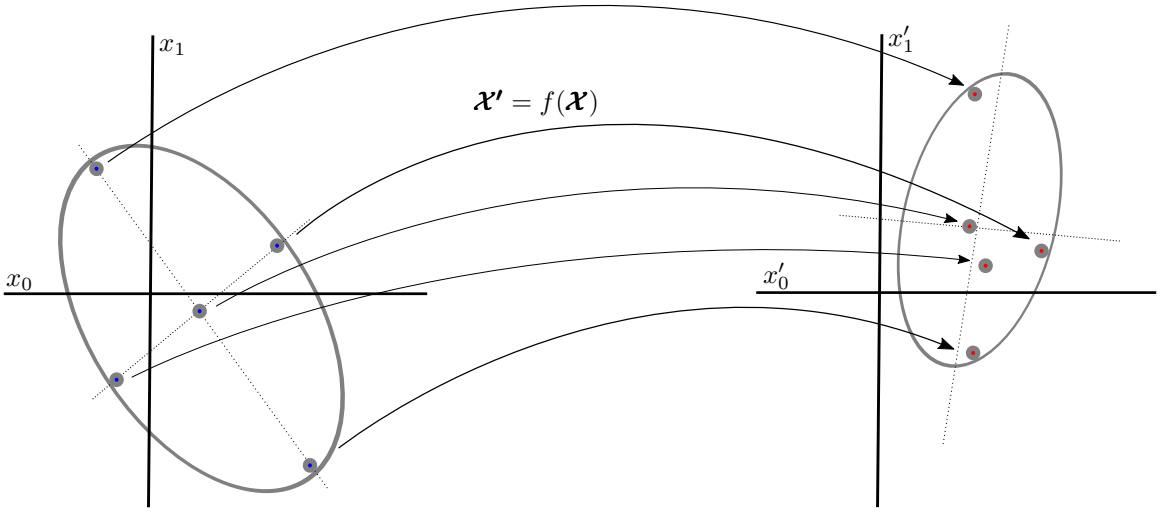


Figure 6.5: A visualisation of how the unscented transform selects a set of sigma points from the mean and covariance of a two dimensional probability distribution. The mean is always at the centre of the dotted axis for the two-dimensional Gaussian. The sigma points are then propagated through a non-linear function and used to calculate a new mean and covariance for the propagated probability distribution.

$$\boldsymbol{M} = \sqrt{\boldsymbol{\Sigma}} \quad \text{if} \quad \boldsymbol{\Sigma} = \boldsymbol{M}\boldsymbol{M}^T, \tag{6.16}$$

$\boldsymbol{M}\boldsymbol{M}^T$ is a matrix product. Numerical methods have been designed which can calculate the square root of a matrix, such as the Cholesky decomposition. From Equation 6.13 it is clear that the first sigma point $\boldsymbol{\mathcal{X}}^{[0]}$ is always the mean of the probability distribution. Each of the selected sigma points have two weights associated with it, one weight w_m is used to compute the mean, while the other weight w_c is used to compute the covariance of the propagated state's probability distributions. The weights are constants, calculated by

$$w_m = \lambda/(n+\lambda) \qquad \text{and} \tag{6.17}$$

$$w_c = \lambda/(n+\lambda) + (1-\alpha^2+\beta), \tag{6.18}$$

where λ, α and β are constants determined by certain knowledge of the underlying Gaussian representation of the probability distributions. After the sigma point transformation, the mean of the transformed distribution can be approximated by calculating the weighted sum of each of the $2n$ sigma points,

$$\boldsymbol{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \boldsymbol{\mathcal{X}}_t^{[i]}. \tag{6.19}$$

The covariance matrix for the transformed distribution can also be calculated,

$$\boldsymbol{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\boldsymbol{\mathcal{X}}_t^{[i]} - \boldsymbol{\mu}_t)(\boldsymbol{\mathcal{X}}_t^{[i]} - \boldsymbol{\mu}_t)^T, \tag{6.20}$$

which results in an approximated Gaussian probability distribution of sigma points that were propagated through a non-linear function.

6.2.2 The UKF Algorithm

The UKF algorithm looks very complicated on the surface, however with careful consideration of each equation, it becomes a lot easier to digest. Table 6.1 summarises the entire algorithm, and hereafter follows a detailed explanation of the inner workings of the UKF. The algorithm starts with the prediction step by selecting sigma points from the belief distribution of all of the states $bel(\mathbf{x}_{t-1})$ at the previous time step. The sigma point matrix consists of $2n+1$ rows and columns, and is calculated according to the rules in Equations 6.13 - 6.15. The first term in the brackets denotes the first row of sigma points in the matrix, the second term denotes rows 1 to n and the last term denotes rows $n+1$ to $2n$,

$$\boldsymbol{\mathcal{X}}_{t-1} = \begin{pmatrix} \boldsymbol{\mu}_{t-1} & \boldsymbol{\mu}_{t-1} + \gamma\sqrt{\boldsymbol{\Sigma}_{t-1}} & \boldsymbol{\mu}_{t-1} - \gamma\sqrt{\boldsymbol{\Sigma}_{t-1}} \end{pmatrix}. \tag{6.21}$$

All of the sigma points are then propagated through a motion model $\boldsymbol{g}$ given a control input $\mathbf{u}_t$, to calculate the predicted sigma points. The sigma points are propagated by means of the unscented transform,

$$\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} = \boldsymbol{g}(\boldsymbol{\mathcal{X}}_{t-1}^{[i]}, \mathbf{u}_t) \qquad \text{for } i = 0...2n, \tag{6.22}$$

which estimates the predicted belief distribution $\overline{bel}(\mathbf{x}_t)$ by approximating its mean and covariance according to Equations 6.19 - 6.20. The mean of the predicted belief distribution is calculated from the propagated sigma points by using the weight constant w_m (calculated in Equation 6.17),

$$\bar{\boldsymbol{\mu}}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\boldsymbol{\mathcal{X}}}_t^{[i]}. \tag{6.23}$$

The covariance of the predicted belief distribution is calculated from the propagated sigma points and the weight constant w_c (calculated in Equation 6.18),

$$\bar{\boldsymbol{\Sigma}}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)(\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)^T + \boldsymbol{Q}_p. \tag{6.24}$$

Here, $\boldsymbol{Q}_{p,t}$ is a process noise covariance matrix which is additively introduced into the system. Hereafter follows the measurement update step, which also starts by selecting sigma points, but from the belief distribution of the current predicted states, using the current predicted covariance and mean,

$$\bar{\boldsymbol{\mathcal{X}}}_t = \begin{pmatrix} \bar{\boldsymbol{\mu}}_t & \bar{\boldsymbol{\mu}}_t + \gamma\sqrt{\bar{\boldsymbol{\Sigma}}_t} & \bar{\boldsymbol{\mu}}_t - \gamma\sqrt{\bar{\boldsymbol{\Sigma}}_t} \end{pmatrix}. \tag{6.25}$$

The sigma points from the predicted belief over the states are propagated through a measurement model $\boldsymbol{h}$,

$$\bar{\boldsymbol{\mathcal{Z}}}_t^{[i]} = \boldsymbol{h}(\bar{\boldsymbol{\mathcal{X}}}_t^{[i]}) \qquad \text{for } i = 0...2n, \tag{6.26}$$

which yields the sigma points of the measurements that would result from the predicted belief over the states. The mean of the belief over the predicted measurements $\hat{\boldsymbol{z}}_t$ is approximated by making use of the weight constant w_m

$$\hat{\boldsymbol{z}}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\boldsymbol{Z}}_t^{[i]}. \tag{6.27}$$

The measurement uncertainty $\boldsymbol{S}_t$ describes the covariance of the predicted measurement belief,

$$\boldsymbol{S}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\boldsymbol{Z}}_t^{[i]} - \hat{\boldsymbol{z}}_t)(\bar{\boldsymbol{Z}}_t^{[i]} - \hat{\boldsymbol{z}}_t)^T + \boldsymbol{R}_{m,t}, \tag{6.28}$$

where $\boldsymbol{R}_{m,t}$ is an additive measurement noise covariance matrix. Equation 6.27 and 6.28 together form the belief distribution over the predicted measurements. The cross-covariance matrix $\bar{\boldsymbol{\Sigma}}^{x,z}$ is calculated by a function that uses the predicted estimates of the states and measurements to determine the covariance of one process with another at certain points in time. The cross-covariance matrix essentially describes how strong the correlation between the measurements and the states are,

$$\bar{\boldsymbol{\Sigma}}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\boldsymbol{X}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)(\bar{\boldsymbol{Z}}_t^{[i]} - \hat{\boldsymbol{z}}_t)^T. \tag{6.29}$$

The Kalman gain is calculated by multiplying the cross-covariance matrix with the inverse of the measurement uncertainty

$$\boldsymbol{K}_t = \bar{\boldsymbol{\Sigma}}_t^{x,z} \boldsymbol{S}_t^{-1}. \tag{6.30}$$

Finally, the mean estimates of the current states are determined by adjusting the predicted estimates with the product of the Kalman gain and the difference between the actual measurements $\boldsymbol{z}_t$ and predicted measurements $\hat{\boldsymbol{z}}_t$

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t). \tag{6.31}$$

The actual measurements are obtained from the output of the image processing algorithm, which gives the positions of landmarks relative to the multi-rotor. Lastly, the current covariance matrix is calculated from the predicted covariance matrix by subtracting a product of the Kalman gain and measurement uncertainty

$$\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t \boldsymbol{S}_t \boldsymbol{K}_t^T. \tag{6.32}$$

The estimates of the current state's mean and the covariance matrix is returned as input for the next iteration step in the UKF algorithm. The entire UKF algorithm is summarised in Table 6.1, where Equations 1-4 perform the prediction step, and Equations 5-12 perform the measurement update step.

Table 6.1: The UKF algorithm

Prediction step:

1. $\boldsymbol{\mathcal{X}}_{t-1} = \begin{pmatrix} \boldsymbol{\mu}_{t-1} & \boldsymbol{\mu}_{t-1} + \gamma\sqrt{\boldsymbol{\Sigma}_{t-1}} & \boldsymbol{\mu}_{t-1} - \gamma\sqrt{\boldsymbol{\Sigma}_{t-1}} \end{pmatrix}$

2. $\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} = \boldsymbol{g}(\boldsymbol{\mathcal{X}}_{t-1}^{[i]}, \mathbf{u}_t) \qquad \text{for } i = 0...2n$

3. $\bar{\boldsymbol{\mu}}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\boldsymbol{\mathcal{X}}}_t^{[i]}$

4. $\bar{\boldsymbol{\Sigma}}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)(\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)^T + \boldsymbol{Q}_p$

Measurement update step:

5. $\bar{\boldsymbol{\mathcal{X}}}_t = \begin{pmatrix} \bar{\boldsymbol{\mu}}_t & \bar{\boldsymbol{\mu}}_t + \gamma\sqrt{\bar{\boldsymbol{\Sigma}}_t} & \bar{\boldsymbol{\mu}}_t - \gamma\sqrt{\bar{\boldsymbol{\Sigma}}_t} \end{pmatrix}$

6. $\bar{\boldsymbol{\mathcal{Z}}}_t^{[i]} = \boldsymbol{h}(\bar{\boldsymbol{\mathcal{X}}}_t^{[i]}) \qquad \text{for } i = 0...2n$

7. $\hat{\boldsymbol{z}}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\boldsymbol{\mathcal{Z}}}_t^{[i]}$

8. $\boldsymbol{S}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\boldsymbol{\mathcal{Z}}}_t^{[i]} - \hat{\boldsymbol{z}}_t)(\bar{\boldsymbol{\mathcal{Z}}}_t^{[i]} - \hat{\boldsymbol{z}}_t)^T + \boldsymbol{R}_{m,t}$

9. $\bar{\boldsymbol{\Sigma}}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\boldsymbol{\mathcal{X}}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)(\bar{\boldsymbol{\mathcal{Z}}}_t^{[i]} - \hat{\boldsymbol{z}}_t)^T$

10. $\boldsymbol{K}_t = \bar{\boldsymbol{\Sigma}}_t^{x,z} \boldsymbol{S}_t^{-1}$

11. $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t)$

12. $\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t \boldsymbol{S}_t \boldsymbol{K}_t^T$

return $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$

6.3 Implementation

The algorithm discussed thus far is for the generic UKF, however its implementation can vary depending on the application and system characteristics. The structure of the UKF used in this localisation problem is illustrated in Figure 6.6, which highlights some of the differences to the standard algorithm.

In any UKF application, the description of the state propagation functions and their inputs are unique to that system. It is therefore very important to clearly define these functions and their parameters. The motion model $\boldsymbol{g}$ that performs the state propagation (step 2 in Table 6.1) is defined in §4.2, where it takes the previous state belief and a vector $\mathbf{u}_t$ as input parameters. In the general UKF, $\mathbf{u}_t$ is defined as the control input vector; however, in this thesis there is no control input,

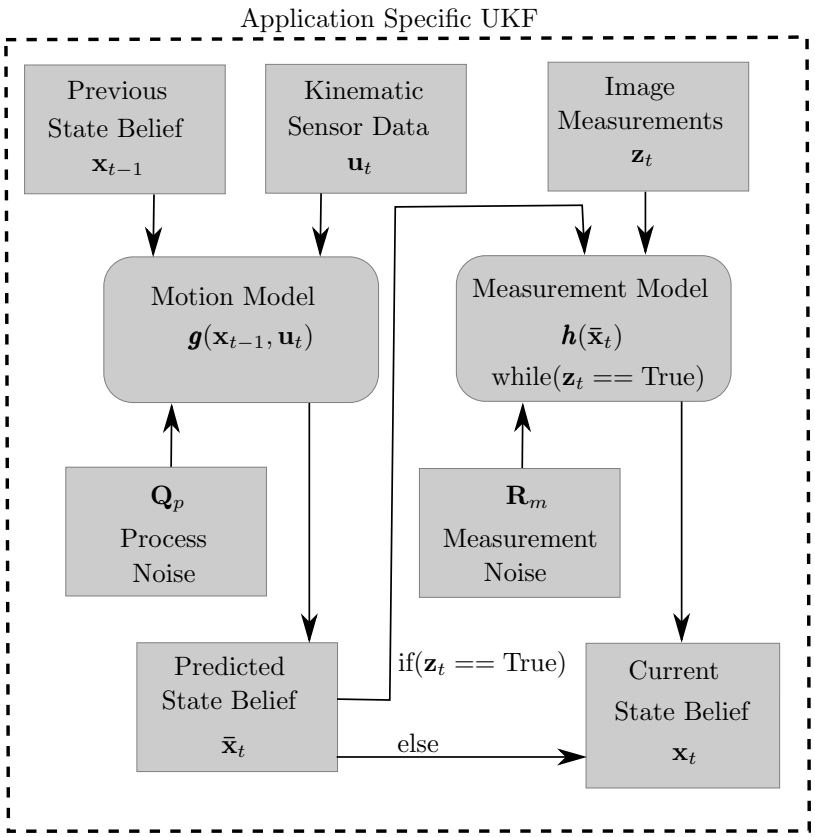


Figure 6.6: A flow chart of the UKF structure as implemented in this localisation problem.

---

only a kinematic observation of sensors, whose values are still represented by $\mathbf{u}_t$. The non-linear transform that is performed by $\boldsymbol{g}$ results in a vector expressed as

$$\boldsymbol{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) => [X_t \quad Y_t \quad Z_t \quad \phi_t \quad \theta_t \quad \psi_t \quad O_{LA}]. \tag{6.33}$$

The result of the transformation performed by $\boldsymbol{g}$ is an update of each of the seven states based on the available kinematic information. The seven states include the three translation variables relative to the landmark reference frame, three Euler angle rotations in the landmark reference frame and the landmark alignment angle. The measurement model $\boldsymbol{h}$ that performs the measurement prediction (step 6 in Table 6.1) can be solved according to the procedure that is defined in §4.3, and the resulting vector of range, azimuth and elevation values is expressed as

$$\boldsymbol{h}(\mathbf{x}_t) => [R_1 \quad \alpha_1 \quad \beta_1 \quad R_2 \quad \alpha_2 \quad \beta_2 \quad ... \quad R_i \quad \alpha_i \quad \beta_i], \tag{6.34}$$

where i is the number of visible landmarks. The noise values that are added to each term of the resulting vector is drawn from a normal distribution described by a certain covariance and a mean of zero. The process noise is described by the process noise covariance matrix (reproduced for clarity)

$$\mathbf{Q}_p = \begin{bmatrix} \sigma_X^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_Y^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_Z^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\psi^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{O_{LA}}^2 \end{bmatrix}, \tag{6.35}$$

and the measurement noise is described by the measurement noise covariance matrix

$$\mathbf{R_m} = \begin{bmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 \\ 0 & 0 & \sigma_\beta^2 \end{bmatrix}. \tag{6.36}$$

The sigma points that are propagated through the non-linear motion and measurement functions are calculated by Equations 6.13 - 6.15, where the total number of sigma points is $2n+1$. There are exactly seven states in this localisation problem, and therefore a total of 15 sigma points are chosen for each state. The λ, α and β parameters that are used in the calculation of the weights (Equations 6.17 - 6.18) are all constants determined by underlying knowledge of the Gaussian distributions that represent the states. In this thesis, α is chosen as 1.08, and β is chosen as 2, such that λ can be calculated by

$$\lambda = \alpha^2(n + \kappa) - n, \tag{6.37}$$

where κ is chosen as 0 (since the PDFs represented by the sigma points are assumed to be Gaussian), and λ is calculated to be 1.168.

The nature of the landmarks used in this thesis allows for multiple landmark measurements to be visible in a single image; however, only one set of kinematic sensor data is available for each set of measurements. This requires more than one measurement update (one for each visible landmark) for every motion model update (prediction step). The UKF is designed to estimate states to the best of its ability with the available information, meaning that it can still estimate the states of the multi-rotor, even if no more landmarks are in sight. The algorithm therefore checks if landmark measurements are available for each iteration, and if not, it will only perform the motion model update, since kinematic data will be available in every iteration (illustrated in Figure 6.6). However, estimating states through the prediction step only will cause an increase in the covariance matrix, since the confidence in its state estimates decrease. This behaviour is illustrated in Figure 6.8 where the confidence ellipse, which indicates the expected range of the belief state, grows when no landmarks are visible.

One of the prerequisites for the UKF to quickly converge on an accurate belief of the states, is that of an initial state estimate. Given enough sensor information, the UKF will often converge to an accurate belief of the states over time; however, if the initial state inputs are a close reflection of reality the filter will converge much faster. This is often a challenge in global localisation problems; however, in this thesis the purpose of the UKF is to localise the multi-rotor relative to artificially placed landmarks with known positions. The UKF algorithm is modified to only start localisation once three or more landmarks are visible, at which point there are enough measurements to calculate the pose of the multi-rotor from the measurements only.

The pose is calculated by making use of the landmark measurements; an axes transformation (Equation 6.38) is performed on the landmark measurements which calculates the relative position of the multi-rotor to the landmarks, given the orientation of the camera that took the images. A vector $\mathbf{p}_B$ exists for every visible landmark, which describes the position of the landmark relative to the camera from a certain orientation,

$$\mathbf{p}_L = \mathbf{R}_{(\psi)}\mathbf{R}_{(\theta)}\mathbf{R}_{(\phi)}\mathbf{p}_B, \tag{6.38}$$

where $\mathbf{p}_L$ is a vector describing the position of the multi-rotor in the landmark axes. The yaw, pitch and roll angles used in this axes transformation is equal to the orientation of the multi-rotor, measured in the NED axes system. This axes transformation is similar to Equation 4.7, where it takes image sensor measurements in the body-fixed reference frame and transforms them to the landmark reference frame. However additional information is required for the initial estimation of the landmark alignment angle state, which is acquired through an image processing technique. As soon as two even numbered landmarks are detected, a line is drawn between their reference points (the white line between landmark zero and two in Figure 6.7).

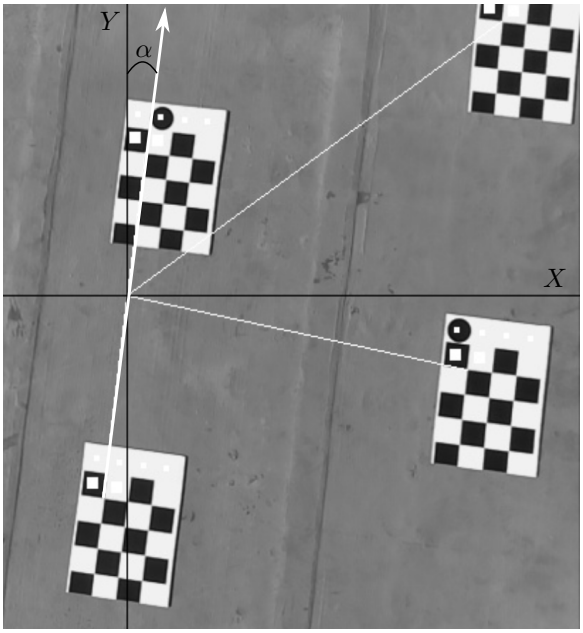


Figure 6.7: Calculation of the angle of inclination α.

The gradient of that line is used to calculate the angle of inclination with the image Y-axis, which determines the yaw angle of the camera relative to the landmark Y-axis (after accounting for pitch and roll through an Euler rotation transformation). Using the relative yaw information along with the compass heading measurement it is possible to calculate the initial landmark alignment angle, which will converge to a constant value through the UKF state estimation. Furthermore, state estimation of angles can often cause problems owing to the nature of angle measurements. For example, a positive 365 degree angle around the yaw axis of the multi-rotor is equal to a positive five degree angle of rotation; therefore, when calculating the mean of a few angles (such as a set of sigma points) with normal algebra, the result would be incorrect. To calculate the mean of several

angles correctly, simple normalisation functions are used to correct for any peculiarities that may occur in calculation of the mean or residual of angles.

Various changes were made to the general UKF algorithm in order to implement a functional state estimator for the problem defined in this thesis. Several additional functions were developed which utilises unique methods to solve for variables to enable the localisation algorithm to perform successfully. Any facet of the UKF's implementation that is not explicitly discussed in this section was implemented according to the standard definition of the UKF as discussed in §6.2.2 and defined in Thrun [26].

6.4 Simulation

To verify that the UKF localisation algorithm functions correctly, it was first tested in simulation, which provides a controlled environment with artificially generated sensor measurements and sensor noise. The sensor measurements were calculated from the multi-rotor states generated in simulation, and are also subject to the same noise models that are used in the UKF. The multi-rotor states that are generated in simulation are used as a ground truth with which to compare the estimated states, thereby testing the accuracy and functionality of the UKF in simulation before practical tests are conducted.

The purpose of the simulation is to be a realistic representation of all the aspects of a real-world test flight and to generate all of the flight data accordingly. However, a simulated test flight will never be a true representation of state propagations when compared to a similar practical test flight. A fixed-wing aircraft has a specific flight envelope and a very detailed motion model which defines how states progress from one time step to the next. A multi-rotor, however, is a unique type of aircraft owing to its agility. This means that there is no previous state in orientation or velocity that constrains a certain next-state value. Therefore, any movement that occurs in the simulated flight has the possibility of occurring in a real flight, however unlikely. Therefore, it is not critically important that the simulated flight path and states of the multi-rotor resembles an actual multi-rotor's behaviour, and any moderately realistic progression of states will be sufficient.

The states that are generated in simulation is representative of the ground truth; however, in reality there is no way of truly knowing the ground truth. It is therefore necessary to add noise to sensor measurements that are derived from the ground truth state values. The noise that is added to velocity and orientation measurements which are calculated from the states is drawn from the same normal distribution that characterises the process noise in §4.2. Another important effect that needs to be considered in simulation is that of landmarks coming into and out of view as the multi-rotor flies over them. This happens in practical tests owing to the field of view of the camera; however, if unaccounted for in simulation, all the landmarks will be visible at all times. This was accounted for by defining a virtual field of view whereby only landmarks with certain positions are used to generate measurements, thereby representing a real-world scenario accurately.

The ground truth used in this thesis was designed to simulate a flight where various different combinations of all the states are present at some point in time. In the simulated flight, the multi-rotor flies up to an altitude of three meters, after which it flies in a snake-like pattern over the staggered landmarks (visible in Figure 6.8). The orientation of the multi-rotor also changes according to the velocity vector to mimic realistic flight characteristics. The data was simulated at a sample rate of five hertz, which resembles the sample rate of the data acquisition system mounted on the multi-rotor. The localisation algorithm was tested against the simulated flight data to isolate

any unexpected behaviour and to observe if the estimated states are close the ground truth, at which point accurate localisation is performed. A useful metric which can be used to determine the accuracy of localisation is a confidence ellipse, which visually illustrates the state belief as a function of the covariance that characterises the Gaussian belief distributions over the states. A three-sigma confidence ellipse illustrates the region wherein all the state values will be that fall within three standard deviations of the mean of the belief distribution.

The results of the state estimation on the simulated data set is shown in Figures 6.8 and 6.9, and the position errors are plotted in Figures 6.10 and 6.11, where each point represents the deviation from the ground truth position as time progresses. The error drastically increases when no more landmarks are visible, this is also illustrated by the increase in the confidence ellipse in Figures 6.8 and 6.9, which also grows near the end of the simulated flight when no landmarks are visible. The mean position errors measured from the ground truth in the simulated state estimation is summarised in Table 6.2. The mean error in the XYZ landmark axes is calculated from position errors only whilst landmarks are visible.

Table 6.2: Simulation mean position and orientation errors.

X-error	Y-error	Z-error	Yaw Error	Pitch Error	Roll Error
2.1 cm	4.6 cm	5 cm	0.15^o	0.23^o	0.19^o

The results presented in Table 6.2 and Figures 6.8 - 6.10 illustrate the localisation algorithm performing successfully in the pose estimation of the multi-rotor. From the figures it is clear that the confidence ellipses shrink smaller as landmarks become visible, and increase when landmarks are no longer visible. The grey bars in the error plots indicate the three-sigma confidence range of the position error according to the covariance matrix at each time step. The ground truth orientation used in the simulation is plotted in Figure 6.11b, where the angles used in the UKF algorithm is subject to the same noise that was modelled in §4.2.

The simulation environment and artificial data sets proved to be extremely useful in the design, testing and fault diagnosis of the localisation algorithm. The simulation undoubtedly saved a lot of time which otherwise would have been lost on unsuccessful experiments, had several functional problems in the algorithm not been solved before flight tests were conducted. The simulation provided an indication of what can be expected from results of the experimental flight tests, in a best case scenario.

6.5 Summary

The principle of state estimation through Bayesian filtering was briefly discussed, followed by an overview of the theory behind Gaussian distributions. The concepts upon which the UKF are based were explained, and the algorithm analysed in a step-wise manner. The application specific implementation of the UKF was detailed, followed by an analysis of the localisation algorithm's performance in simulated flight tests. The algorithm localised the multi-rotor at a mean accuracy of four centimetres relative to the generated ground truth. Furthermore, the algorithm estimated the multi-rotor's orientation at a mean accuracy of 0.2 degrees, which provides a rough indication of the accuracy that can be expected from orientation estimates in actual flight tests, since no baseline or ground truth will be available for attitude comparison.

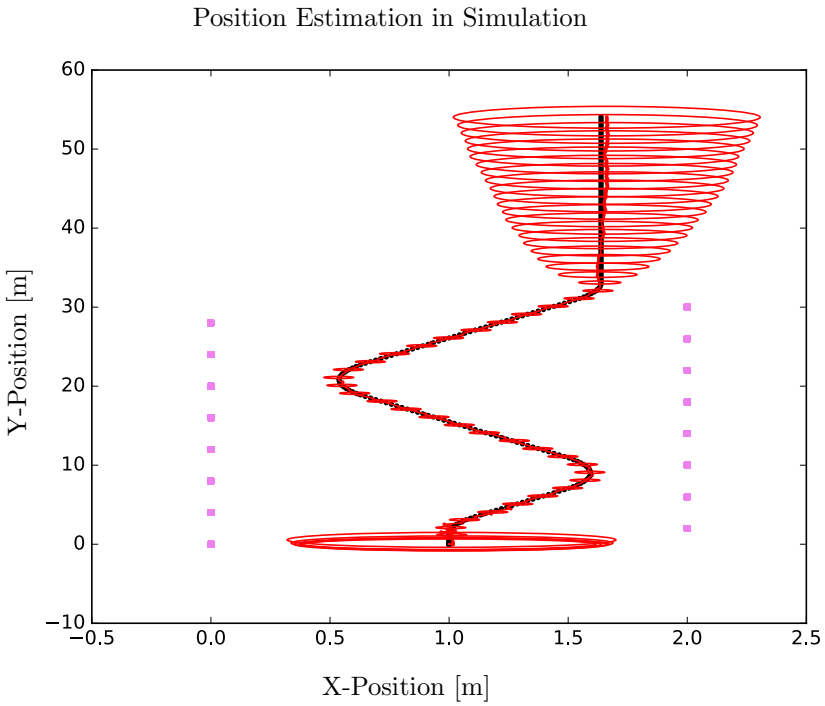


Figure 6.8: The three-sigma confidence ellipses visually illustrating the covariance of the estimated states in the XY landmark axes, where the squares represent the landmarks.

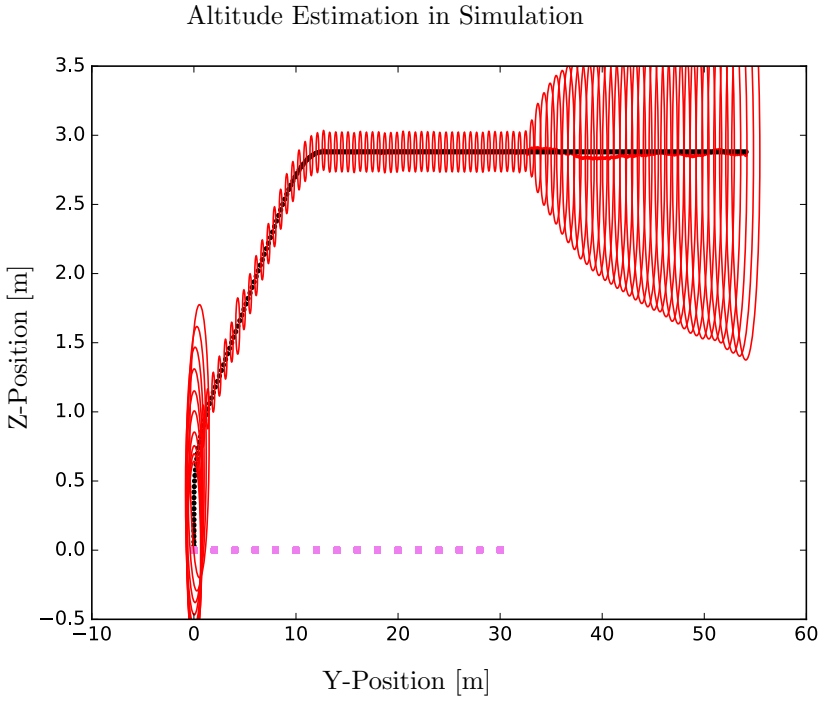
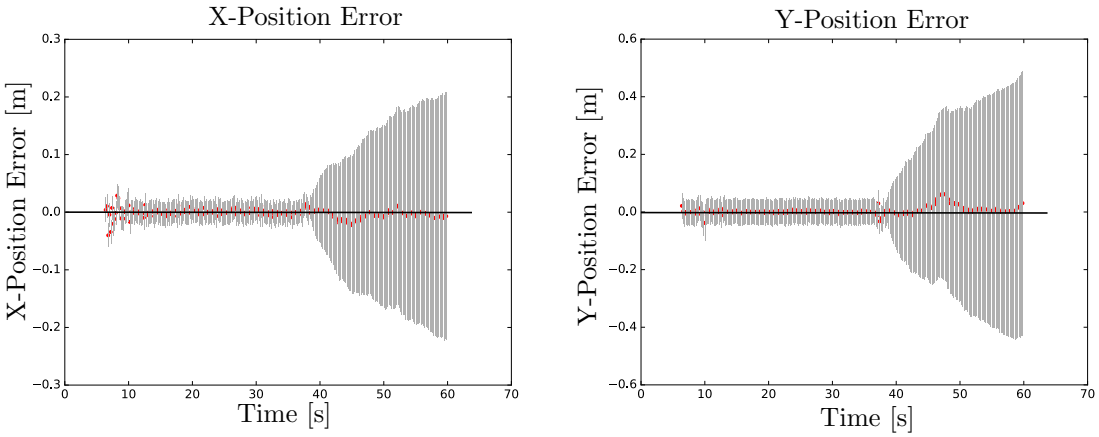


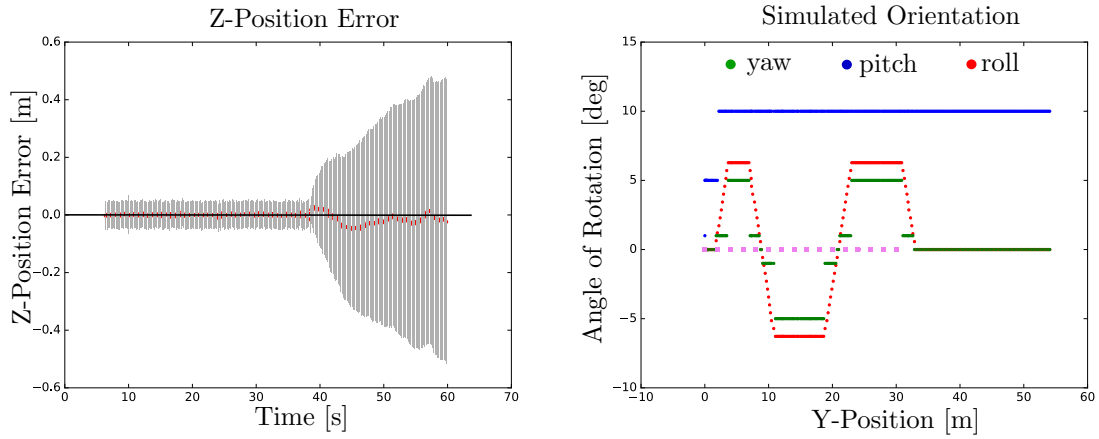
Figure 6.9: The three-sigma confidence ellipses visually illustrating the covariance of the estimated altitude state. The confidence ellipses demonstrate the same behaviour as the XY position estimates.



(a) X-position error in simulation.

(b) Y-position error in simulation.

Figure 6.10: (a): The X-axis position error in simulation. (b): The Y-axis position error in simulation. Each point represents the deviation from the ground truth, and the grey bars represent the three-sigma confidence range of the position error according to the covariance matrix.



(a) Z-position error in simulation.

(b) Simulated orientation

Figure 6.11: (a): The Z-axis (altitude) estimation error in simulation. (b): The ground truth simulated orientation plotted against the landmark Y-axis.

Chapter 7

Experimental Results

Successful state estimation is obtained by the implementation of several subsystems which need to work in unison. In this project, the desired purpose of the state estimator is to perform accurate localisation through monocular vision. This chapter discusses the experimental test procedure and analyses the localisation results obtained from practical flight test data. The differential GPS system is discussed and evaluated in its reliability as a ground truth measurement to which the localisation estimates are compared. Finally, a conclusion is drawn with regards to the accuracy of the localisation algorithm designed in this thesis and its applicability in antenna array characterisation applications.

7.1 Experimental Test Design

The experimental test design and set-up is especially important in a localisation application where landmarks are manually placed in the environment. This is owing to the assumption that all landmark positions are known accurately, which in turn requires accurate placement of the landmarks. The localisation algorithm predicts a landmark's position in the measurement update and compares it to the actual measured position. The algorithm then adjusts the state estimates to attempt a better prediction in the next time step. However, if the landmark's actual position is different to the localisation algorithm's internal knowledge, the predicted landmark positions will be completely inaccurate and the localisation algorithm will not converge to an acceptable estimate of the states. It is therefore important that landmarks are placed at the exact positions as specified in the localisation algorithm.

Another important aspect in this localisation application is the requirement of three or more visible landmarks for the algorithm to converge on an accurate estimate of the states; this is ensured by using a staggered landmark pattern. The required distance between landmarks are determined by the field of view of the camera so that three or more landmarks are visible at an altitude of roughly three meters. The staggered pattern of landmarks ensures that, whilst the multi-rotor is in forward flight, as one landmark goes out of the field of view, another landmark simultaneously comes into the field of view of the camera. The dimensions and landmark pattern is visually illustrated in Figure 7.1.

The exact effect of landmark placement accuracy on the performance of the localisation algorithm is unknown. Whilst very accurate placement of landmarks is preferable, there is a certain point at which extreme placement accuracy becomes impractical to implement whilst having no added benefit

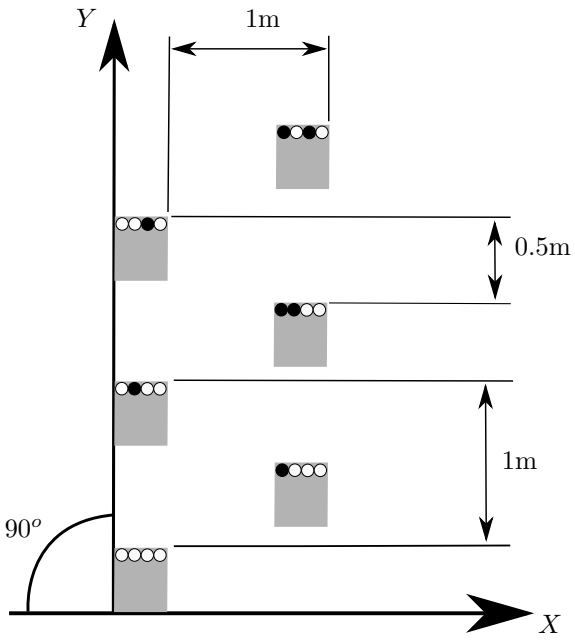


Figure 7.1: Landmark placement pattern and dimensions, demonstrating how each unique binary numbered landmark fits into the pattern.

---

to the localisation algorithm. The mean errors for the algorithm in simulation are summarised in Table 6.2, which represents the best case localisation accuracy in a controlled environment with perfect landmark placement. The requirement for landmark placement accuracy to ensure adequate performance of the localisation algorithm is not explicitly defined. Therefore, out of necessity, it is assumed that the most accurate replication of the landmark placement in simulation is sufficient and that small variations (measured to be under 0.5 centimetres) will not impact the localisation algorithm in any adverse way. To ensure the best possible placement accuracy, each landmark was individually measured relative to one another and relative to the landmark axes's point of origin. The location where the tests were conducted was situated on a flat, concrete surface which ensured accurate placement in the Z-axis as well.

Owing to the nature of the experimental tests, which were conducted on a flying multi-rotor, the weather conditions and environment also played an important role. Therefore, the tests were conducted on calm, sunny days to allow smooth and stable flight over the landmarks. In order to allow optimal performance of the on-board GPS as well as the Piksi DGPS (discussed in §7.3), an elevated and open area was chosen to allow full view of the sky for maximum satellite connectivity.

7.2 Localisation Results

The multi-rotor data acquisition system and the baseline DGPS was used to conduct multiple flight tests, each of which consisted of a single pass over the landmarks. Each flight over the landmarks varied in position, altitude, velocity and orientation; therefore, each data set is significantly different from the others. The Piksi DGPS system was used to measure the position of the multi-rotor at each sampling instance to establish a baseline which can be used as ground truth for comparison with the localisation results. The results of one of the successful flight tests are illustrated in Figures

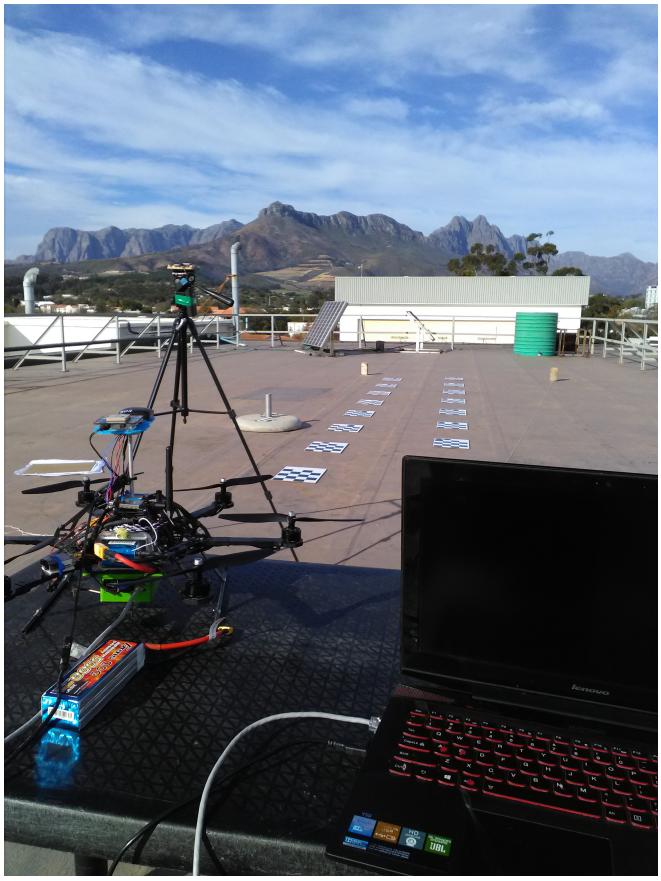


Figure 7.2: The hardware setup that was used to perform the experimental flight tests. The hexacopter platform and DGPS ground station is visible on the left, and the chessboard landmarks are visible in the centre of the photo. The laptop was used to communicate with the on-board computer through ethernet, and execute the python script which performed all of the data capturing.

7.3 to 7.4. The DGPS's real-time kinematic (RTK) measurements are represented by the black dots, and the state estimates are plotted in red (additional localisation results can be found in Appendix A). The ideal experimental flight test would be identical to the simulation, in flight trajectory and landmark placement, since the similarities in the tests would provide confidence in the comparison between practical results and the simulation's results. However, following the exact same flight trajectory as generated in the simulation proved difficult, owing to the lack of precise movement control over the multi-rotor. However, over multiple successful flight tests, several trajectories were executed to collect enough data to determine the mean accuracy of the localisation algorithm when compared to the DGPS's ground truth measurements. The final pose estimation accuracy is derived from a combination of several successful flight tests, which is summarised in Table 7.1.

The localisation algorithm only starts estimating the position of the multi-rotor once three or more landmarks are visible; this allows for a better initial position estimate and faster convergence to an accurate estimate of the states. In general UKF state estimation, more information should give a better state estimate; however, the localisation algorithm requires at least two landmarks in order to estimate the heading of the multi-rotor relative to the landmark axes. If the initial estimate

of the heading is completely wrong, it will have an adverse and possibly unrecoverable effect on further state estimates.

It is clear that the position estimation closely follows the baseline measurements in both XY-position and Z-altitude; however, the estimates drift away from the baseline when landmarks are no longer in sight (clearly visible in the altitude plot in Figure 7.4). This behaviour is expected, because the belief over the states becomes more uncertain due to less available information in the measurement updates. As a result, the mean of the estimates drift; similar behaviour was also visible in simulation, where the estimates drift away from the baseline when no landmarks are visible. The estimated trajectory is plotted to scale on each axis to visually illustrate the localisation accuracy relative to the baseline measurements in the landmark axes.

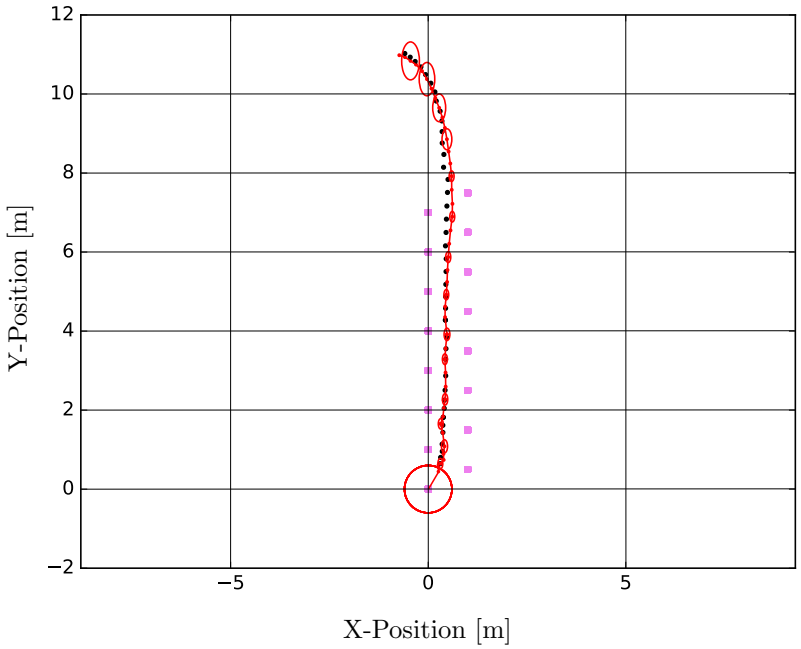


Figure 7.3: The estimated flight trajectory for one of the successful flight tests showing the horizontal movement of the multi-rotor. The pink squares represent the landmark positions, the red dots are the estimated positions and the red lines show the most likely flight trajectory between sample instances. Each sample instance has a corresponding confidence ellipse that is also plotted in red. The black dots represent the DGPS measurements which are plotted as a position baseline. The mean errors are 5.9 cm along the X-axis and 6.2 cm along the Y-axis, relative to the baseline measurements. The ellipses are plotted at every third iteration for visual clarity.

The localisation accuracy is illustrated by means of error and covariance ellipses, each providing an insight into different aspects of the localisation algorithm's performance. The covariance ellipses (plotted in red) in Figures 7.3 and 7.4 visually illustrate the two-dimensional confidence range around the estimated states, and is constructed from the values in the covariance matrix for that sampling instance. The three-sigma confidence ellipses are plotted to scale in order to clearly illustrate the behaviour of the ellipses throughout the test.

The error plots in Figures 7.5a - 7.5c show the difference between the estimate of the state and

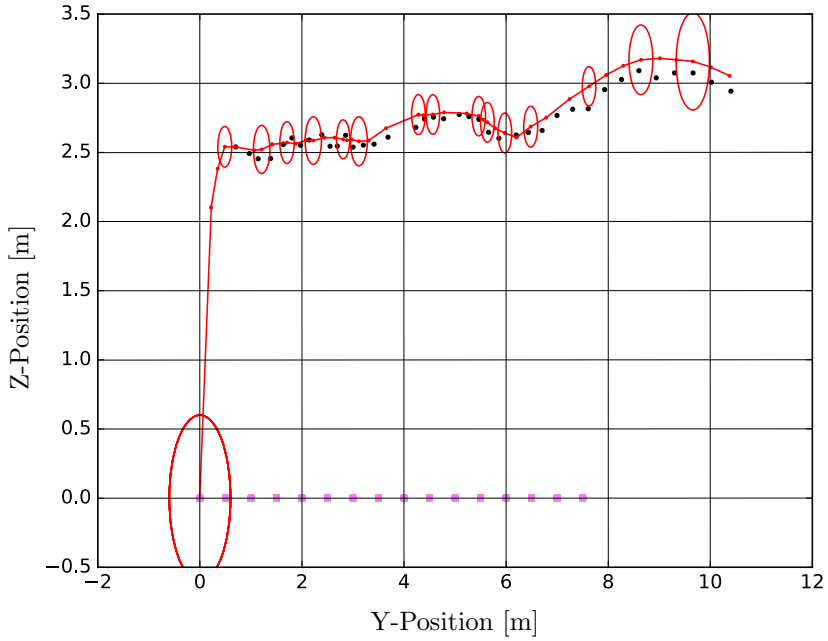
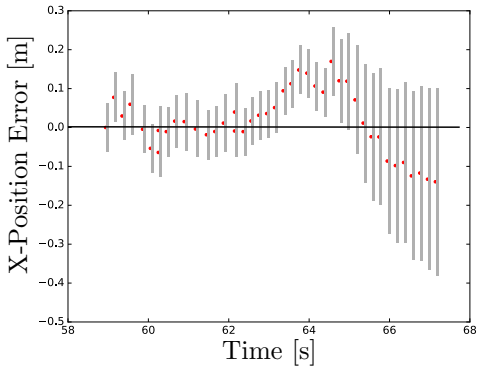


Figure 7.4: The estimated altitude trajectory showing the vertical movement for one of the successful flight tests. The pink squares represent the landmark positions, the red dots are the estimated positions and the red lines show the most likely flight trajectory between sample instances. Each sample instance has a corresponding confidence ellipse that is also plotted in red. The black dots represent the DGPS measurements which are plotted as a position baseline. The mean altitude error along the Z-axis is 7.8 cm compared to the baseline measurements.

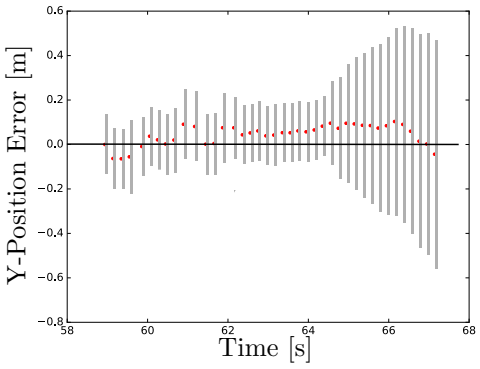
---

the baseline measurement at each sampling instance. A point that is closer to the zero-line indicates a more accurate state estimate, where the grey bars represent the three-sigma confidence range. The confidence range indicates the expected value of a state within three standard deviations of the mean of the state's probability distribution. Therefore, if the grey bar overlaps with the zero-line in the sample instance, the estimation error is small enough so that the estimated state value falls within the three-sigma range of the baseline measurement. As expected, the error and confidence range both increases when no landmarks measurements are available towards the end of the test flight.

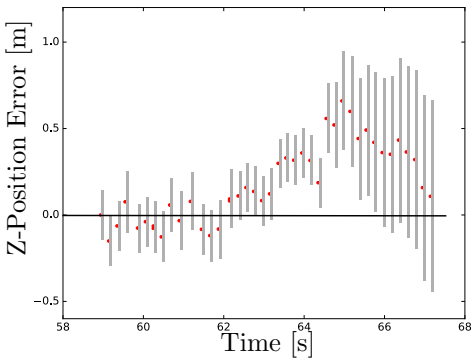
An interesting observation is that of the striking numerical similarity between the final covariance matrix in simulation and the final covariance matrix in several of the flight tests. The final covariance matrix is sampled at the last iteration where landmark measurements were available at the end of the flight, and it is therefore a good indication of the localisation algorithm's confidence in its state estimates. The diagonal elements in the final covariance matrix in both simulation and practical flight data are very similar. The identity between the two covariance matrices is due to the similarities between the simulated flights and the experimental flight tests, as well as similar landmark placement patterns and noise models. This observation is a positive indication that the localisation algorithm functions correctly under the influence of non-ideal effects in flight tests that could not be modelled in simulation. Furthermore, by observing the changes in the covariance matrix by means of confidence



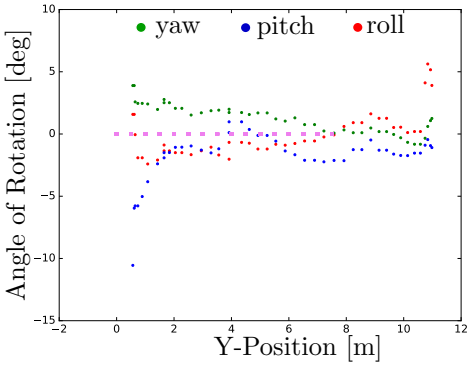
(a) Flight test X-position estimation error.



(b) Flight test Y-position estimation error.



(c) Flight test altitude estimation error.



(d) Flight test orientation.

Figure 7.5: (a)-(c): The flight test position and altitude estimation error plots, which visually illustrate the difference between the estimate of the state and the baseline measurement of the state. The grey bars show the three-sigma confidence range of the state estimates. (d): The estimated orientation of the multi-rotor.

ellipse plots before and after the measurement update (illustrated in Figure 7.6), it is possible to verify that the localisation algorithm's confidence in its state estimates increase significantly after the measurement update.

The localisation accuracy of the UKF state estimation algorithm is quantified by a set of mean position errors and standard deviations, which are summarised in Table 7.1. The orientation of the multi-rotor is estimated from sensor measurements on the flight controller. The standard deviation of the orientation estimates were derived from data sets generated in a simulated real-world scenario, where the multi-rotor motors are powered up and the sensors are exposed to all of the noise sources that would be present in the experimental flight tests. The pose of the multi-rotor was kept constant whilst orientation measurements were taken, and the resulting data set was used to calculate the standard deviation of the angle of rotation around each axis. The resulting noise matrix was used in simulation, and the test results produced a mean error that can be expected in orientation estimates from the localisation algorithm. There was no way of measuring a ground truth of the multi-rotor's orientation without a more accurate and expensive IMU. The expected mean error in orientation

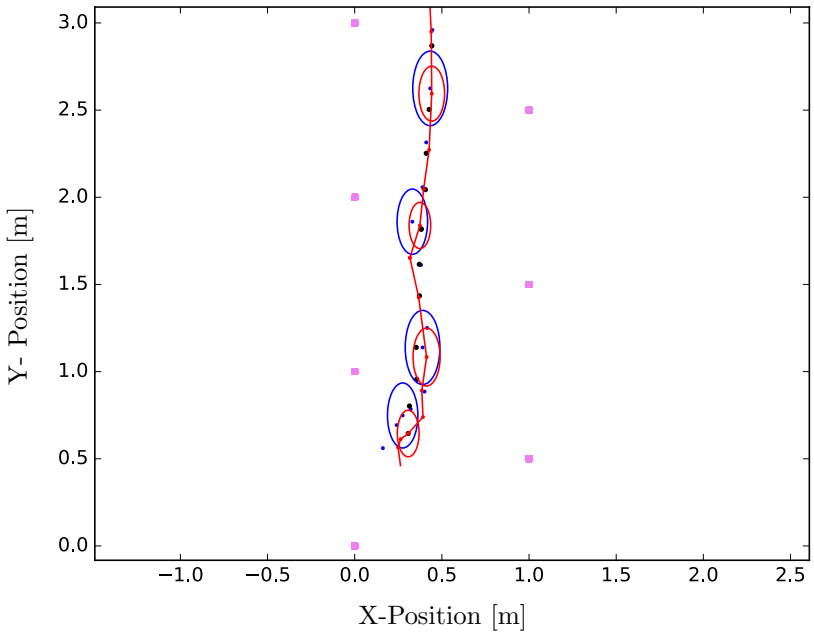


Figure 7.6: The confidence ellipses plotted from the covariance matrix before and after the measurement update. The blue dot and ellipse indicates the mean and its three-sigma confidence before the measurement update, and the red dot and ellipse indicates the mean and three-sigma confidence after the measurement update.

---

measurements are also summarised in Table 7.1.

The position of the multi-rotor was estimated over several flight tests and compared to the baseline along each of the landmark axes. The resulting data sets consisted of 45 data points on average, and were used to compute a mean error in each axis relative to the measured baseline. Furthermore, the covariances for each test was observed to be very similar, and the average of the standard deviations across all tests were calculated for each axis. The mean altitude error in the Z-axis is observed to be larger than the position error in the X- or Y-axis. This behaviour is expected from the results in simulation.

Table 7.1: The mean position errors, mean standard deviation (SD) and expected orientation errors.

X-error	Y-error	Z-error	X-SD	Y-SD	Z-SD	Yaw-error	Pitch-error	Roll-error
6 cm	6.5 cm	7.8 cm	7 cm	13 cm	15 cm	0.15^o	0.23^o	0.19^o

The localisation accuracy in the experimental flight tests is slightly worse than the expected accuracy from the simulation, which was summarised in Table 6.2. The mean errors calculated from the simulated data sets represent a best case scenario in a controlled environment with no unexpected disturbances. However, there are many aspects of a real-world flight test that can simply not be modelled, owing to the unpredictable and erratic behaviour of the disturbances at play. One of the most influential effects not yet considered is that of the baseline measurement noise and accuracy. Manufacturers often claim performance in their products which may be exaggerated

and unobtainable, which has a particularly big influence on the concluding results in this thesis.

7.3 Baseline Reliability

The accuracy results of the localisation algorithm is only as reliable as the baseline to which it is compared. If the baseline measurements are wholly unreliable, it is difficult te determine with certainty the accuracy of the localisation algorithm. The ideal solution for baseline measurements would be a system that can confidently provide millimetre accurate position estimates; however, few of these systems exist. One of the options that do exist is extremely expensive and cannot be implemented in an outdoor environment, which is the Vicon motion capture system [21]. Another possible solution is an optical retro-reflector system that can localise with centimetre accuracy (similar to that of a DGPS); however, no such system was available during the course of this project. Therefore, the only realistic option that was available to measure the localisation accuracy of the algorithm in this project was a low cost DGPS system.

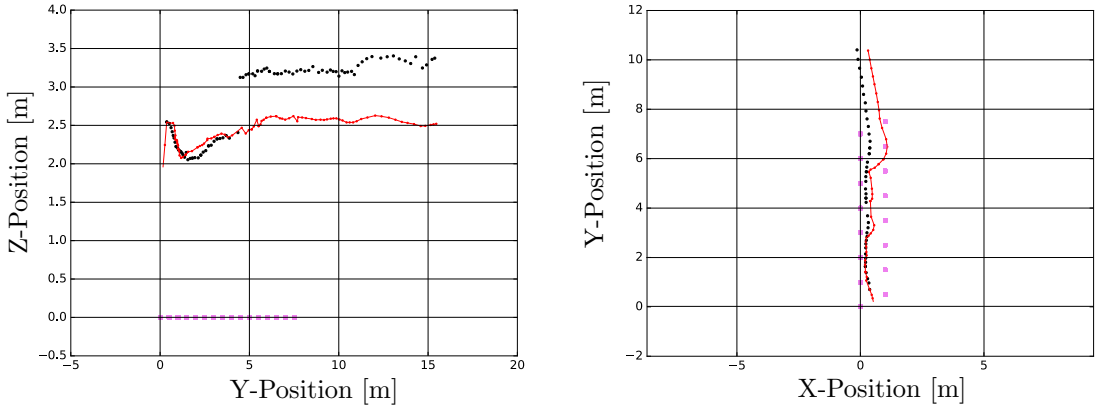
The Piksi DGPS is one of the least expensive RTK systems of its kind and is aimed at the hobbyist market; therefore, its performance is not on par with commercial grade DGPS systems. In order to localise the rover unit that is mounted on the multi-rotor, the DGPS system needs to enter a mode called fixed RTK, at which point it will localise the rover unit with up to two-centimetre accuracy. Fixed RTK is achievable only after a 15-20 minute calibration period; however, very accurate baseline measurements are intermittent, and in some cases easily interrupted by sudden movements of the multi-rotor. Flight tests were unsuccessful more often than not, owing to the unreliability of the Piksi DGPS during the RTK lock process. Therefore, only successful flight tests were used to determine the mean error and overall accuracy of the localisation algorithm. A flight test was considered unsuccessful if the DGPS could not obtain fixed RTK measurements, or if fixed RTK mode was achieved but lost during the flight. These cases were only identified in post processing of the data where the baseline displays unusual behaviour, such as jumping away from the estimated trajectory, as shown in Figure 7.7.

To verify the DGPS noise in a best case scenario, it is possible to measure the standard deviation of static position measurements. To characterise the baseline noise, a similar procedure to that of the measurement noise characterisation was followed, whereby several baseline measurements were captured whilst the multi-rotor was completely stationary. However, it is necessary to note that the DGPS rover unit, whilst in fixed RTK mode, is generally more accurate in its relative position measurements when it is stationary as opposed to when it is moving and changing orientation. The data set should, however, provide some information with regards to optimistic position errors that can be expected in the baseline (summarised in Table 7.2).

Table 7.2: Experimental baseline standard deviations in each of the NED axes.

North	East	Down
1.1 cm	1.1 cm	3.2 cm

There is no way to verify the exact error in the baseline measurements without an even more accurate ground truth measurement. Furthermore, the Piksi DGPS data sheet states that an accuracy of between two and five centimetres are achievable. Assuming that the DGPS performed according to its less accurate specifications, the mean localisation errors summarised in Table 7.1 is most likely not a true or accurate reflection of the actual performance of the localisation algorithm. The true



(a) Vertical baseline drift.

(b) Horizontal baseline drift

Figure 7.7: (a): An example of the DGPS system losing RTK lock during flight, which results in the vertical baseline measurements jumping away from the estimated flight trajectory. (b): Similar behaviour is seen in a different flight test in the horizontal baseline measurements; however, in this case the baseline measurements stagnate and fail to reflect explicit flight trajectory changes by the multi-rotor pilot.

accuracy may be better than the results derived from the baseline comparisons. However, considering that most expensive DGPS systems only achieve up to two centimetre accuracy in perfect implementation, the Piksi DGPS performs well for its price, which is a fraction of commercial grade DGPS systems. Despite the successful flight tests where the baseline looks like it performed well in post-flight data analysis, the Piksi DGPS's performance was not reliable enough to confidently determine if the accuracy of the localisation algorithm is sufficient for application in the MFAA's antenna characterisation.

7.4 Summary

The experimental test set-up consisted of a staggered landmark pattern on a flat surface in an open area. The results of experimental flight tests were presented in the form of position and covariance plots, as well as error plots. The localisation algorithm performed well in several successful flight tests; however, many flight tests were unusable owing to a loss of RTK lock. The resulting accuracy of the localisation algorithm was 6, 6.5 and 7.8 cm along each respective XYZ axis. The expected accuracy in orientation was derived from the simulation, which was 0.15, 0.23 and 0.19 degrees around each respective yaw, pitch and roll axis. Following an analysis of the baseline accuracy, the chapter concludes that the baseline was too unreliable and inaccurate to confidently say whether or not the localisation algorithm is accurate enough for MFAA characterisation.

Chapter 8

Conclusion

This thesis details the development of a monocular vision-based pose estimation algorithm which aims to accurately determine the pose of a multi-rotor UAV system relative to stationary landmarks. Vision-based state estimation is already widely used in ground-based robotic systems; however, there is a lot of potential for research to be done on vision-based localisation of UAVs. As more possible applications of UAVs are realised, the research and development relating to autonomous UAV operations become more relevant and useful. The motivation for the research conducted in this thesis lies in the application of a multi-rotor UAV in radio antenna characterisation. The mid-frequency aperture array (which forms part of the square kilometre array) can utilise a multi-rotor in the characterisation of its antenna clusters, which requires accurate localisation of the multi-rotor. Currently available localisation methods are either cumbersome to implement, very expensive or insufficiently accurate. The primary goal of this project was to solve the problem of accurate localisation whilst minimising the hardware costs.

8.1 Summary

The opening chapter of this thesis introduces the concept of radio antenna arrays and their requirement for characterisation. A multi-rotor is one of the simplest platforms which can be used to characterise antenna arrays, which emphasise the need for accurate localisation thereof. It was found that monocular vision using artificial landmarks is one the most inexpensive methods which can be used to accurately determine the pose of the multi-rotor. A hexacopter with a data acquisition payload was then designed and assembled, making use of inexpensive and open source hardware and software, including a Raspberry Pi on-board computer, Raspicam camera and Pixhawk flight controller.

A low cost system was chosen with the goal of finding out how accurately it can localise, and if it will be sufficient for application in antenna array characterisation. Furthermore, if accurate localisation with an inexpensive vision-based system is proven to be possible but insufficiently accurate, it can be improved by using a higher resolution specialised computer vision camera (based on the image sensor noise discussion that follows after the hardware introduction). A low-cost differential GPS system was available for this project, which was integrated with the on-board computer to record a baseline with which to compare the final localisation results. The multi-rotor system was mathematically modelled in order to sufficiently represent all of the states of interest under the assumption that only system kinematics are considered. A motion- and measurement model

was derived along with their corresponding noise covariance matrices, after which the relevant axes transformations were explained.

The image processing algorithm and consequent landmark design was discussed and motivated. It was decided that a chessboard-type landmark with unique binary patterns is the best choice of landmark to allow for easy detection and recognition in image measurements. The landmarks were placed in exact known locations in a staggered pattern in the environment to insure that a minimum of three landmarks were visible in each image measurement as the multi-rotor flew along an approximate path. The relevant open source image processing functions were detailed, followed by a discussion of the limitations and constraints applicable to image processing in monocular vision applications.

Following a brief study of several filtering techniques in the literature review, the unscented Kalman filter was decided as the best solution to accurately estimate the states of the multi-rotor from kinematic flight data and image measurements. The basic principles relevant to recursive state estimation was discussed, with specific emphasis on the unscented transform, the defining characteristic of the UKF, and the theory thereof. The UKF algorithm was then discussed in detail, explaining how the state vector and covariance matrices are updated for each time step. Following is an explanation of the UKF's implementation in this specific case, where several measurement updates may be required for each motion update, one for each landmark visible in an image measurement. The UKF was first tested in simulation, with promising centimetre-accurate results in position and altitude state estimation. The orientation estimation performed in simulation provided an idea of the accuracy which can be expected from orientation estimates in experimental flight tests, since it was not possible to record a baseline for orientation.

Finally, an experimental flight test procedure was designed to test the accuracy of the localisation algorithm against a baseline. Several flight tests were conducted under ideal conditions to obtain an overall mean error in localisation; however, the baseline measurements proved to be somewhat unreliable, owing to the quality and functionality of the Piksi DGPS system that was used. The final results suggested an average error of seven centimetres in position estimation compared to the baseline, which is larger than expected from the error calculated in simulation. An analysis of the baseline behaviour in practical flight scenarios suggests that the measurements from the Piksi DGPS are too unreliable to confidently determine whether or not the localisation algorithm is accurate enough for the intended application.

The true localisation accuracy of this algorithm may be sufficient for the intended application, but could only be confirmed if a highly accurate baseline measurement method is available, which was not the case during the course of this project. The current suggested localisation accuracy would be suitable and the method very promising for other radio telescope antenna arrays which operate at lower frequencies than the MFAA, like the low-frequency aperture array (LFAA) and hydrogen epoch of reionisation array (HERA). However, in the current state of the project it is not an adequate solution for the purpose of antenna array characterisation for the MFAA frequencies. Despite these results, the localisation algorithm designed in this thesis performed almost on par with a hobby-grade DGPS system at a fraction of the cost.

8.2 Future Work and Improvements

There are several improvements which can be incorporated into the current hardware design that may benefit the localisation accuracy of the system.

1. Make use of a better quality computer-vision camera. The Raspicam is very inexpensive and has very low resolution, by utilising a computer vision specific image sensor with a higher resolution, the noise in the image measurements will be much lower, resulting in better state estimates after the measurement model update.

2. Integrate an inexpensive Piksi DGPS (or similar) into the data acquisition system. The stock GPS module, as with any consumer GPS, is very inaccurate. Incorporating a low quality, hobbyist DGPS that provides position and velocity data that is tenfold more accurate than the stock GPS can also improve the localisation accuracy of the system. Improvements will be made in the motion model state update owing to more accurate velocity information.

3. Measure the true localisation accuracy against a very accurate and reliable baseline. The old variant of the inexpensive Piksi DGPS that was used as a baseline measurement in this project proved to be somewhat unreliable and not nearly as accurate as high-end specialised DGPS systems. A very accurate, true baseline could provide invaluable information regarding the accuracy of the state estimation which can be used to improve the algorithm over several tests. Another solution which can provide an accurate baseline is light-based system which consists of a ground station that tracks a retro-reflector mounted on the multi-rotor with centimetre accuracy. However, the ideal solution to a very accurate baseline is to perform experimental tests in the flying machine arena, which can localise retro-reflecting balls with millimetre accuracy.

Appendices

Appendix A

Additional Results

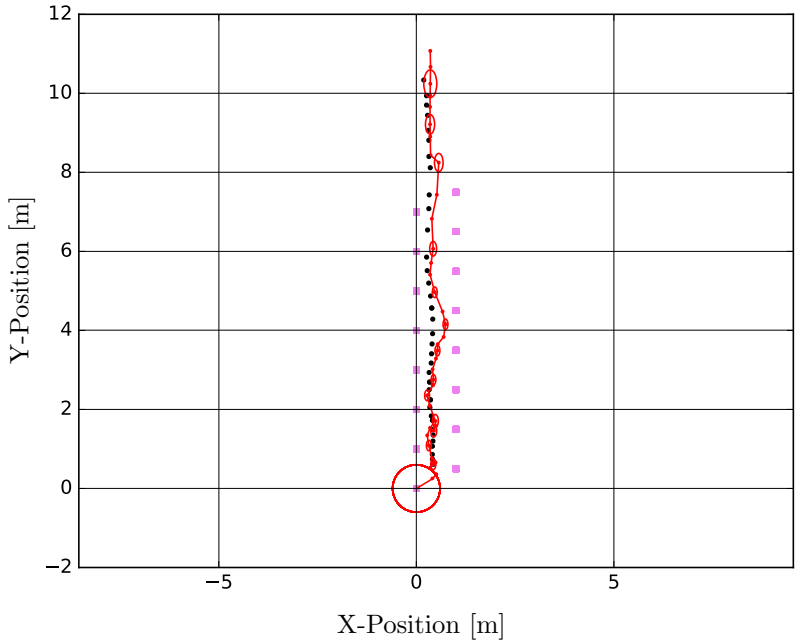


Figure A.1: The estimated flight trajectory in a flight test showing the horizontal movement of the multi-rotor.

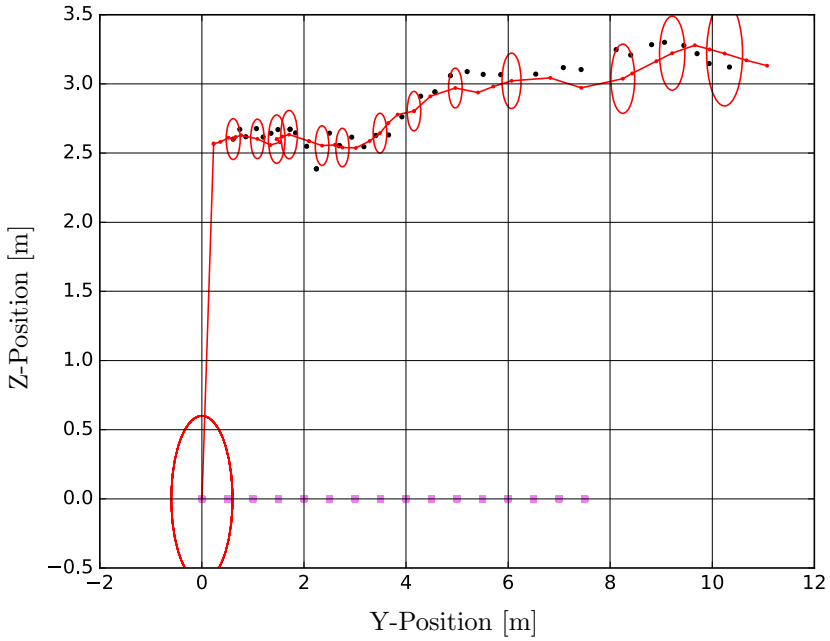


Figure A.2: The estimated altitude trajectory showing the vertical movement in a flight test.

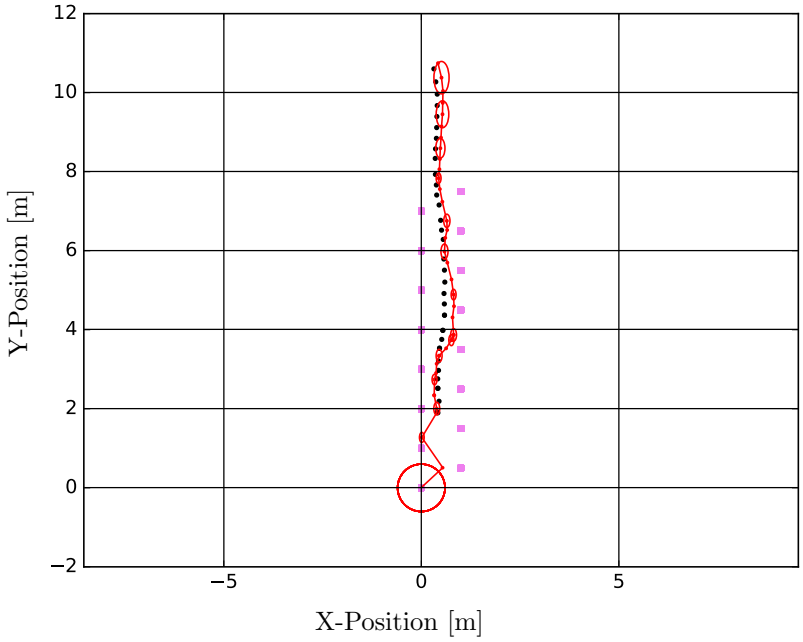


Figure A.3: The estimated flight trajectory in a flight test showing the horizontal movement of the multi-rotor.

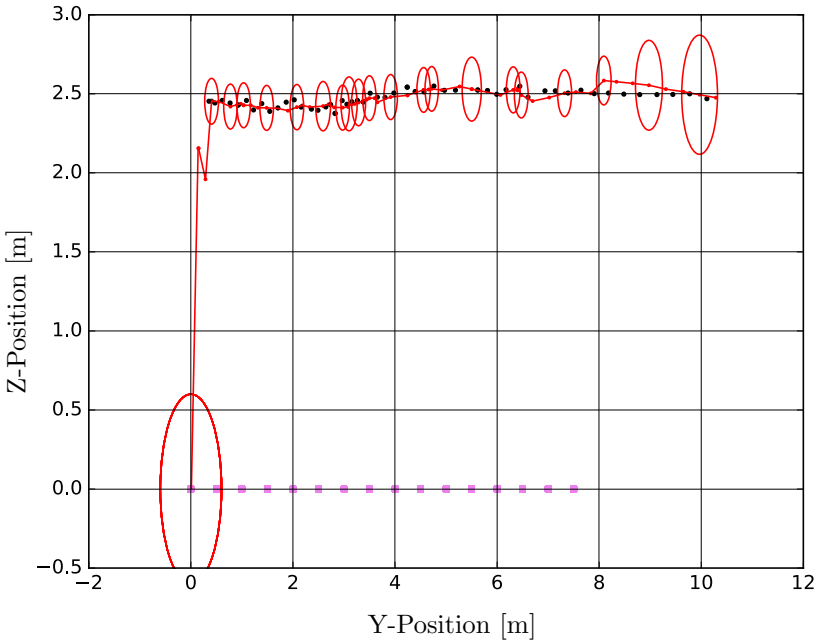


Figure A.4: The estimated altitude trajectory showing the vertical movement in a flight test.

Bibliography

[1] G. Virone, A. M. Lingua, M. Piras, A. Cina, F. Perini, J. Monari, F. Paonessa, O. A. Peverini, G. Addamo, and R. Tascone, "Antenna pattern verification system based on a micro unmanned aerial vehicle (uav)," *IEEE Antennas and Wireless Propagation Letters*, vol. 13, p. 169, 2014.

[2] http://india.skatelescope.org, "Artist impression of mfaa array," 2015. [Online; accessed July 31, 2017].

[3] H. Pienaar and D. Davidson, "Error sensitivity analysis for multi-copter planar positioning on low-gain nearfield measurements," in *Electromagnetics in Advanced Applications (ICEAA), 2016 International Conference on*, pp. 568–571, IEEE, 2016.

[4] G. Virone, F. Paonessa, O. A. Peverini, G. Addamo, R. Orta, R. Tascone, and P. Bolli, "Antenna pattern measurements with a flying far-field source (hexacopter)," in *Antenna Measurements & Applications (CAMA), 2014 IEEE Conference on*, pp. 1–2, IEEE, 2014.

[5] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "Lidar-inertial integration for uav localization and mapping in complex environments," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pp. 649–656, IEEE, 2016.

[6] Mike Tully , "Lidar accuracy specifications," 2012. [Online; accessed September 6, 2017].

[7] A. M. Sabatini, "A stochastic model of the time-of-flight noise in airborne sonar ranging systems," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 44, no. 3, pp. 606–614, 1997.

[8] K. Schreve, "How accurate can a stereovision measurement be?," in *Research and Education in Mechatronics (REM), 2014 15th International Workshop on*, pp. 1–7, IEEE, 2014.

[9] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, pp. 1063–1068, IEEE, 2006.

[10] C. Bellingham, "The accuracy of binocular v monocular vision. a note on apparatus," *The Australasian Journal of Psychology and Philosophy*, vol. 4, no. 4, pp. 301–302, 1926.

[11] L. Jayatilleke and N. Zhang, "Landmark-based localization for unmanned aerial vehicles," in *Systems Conference (SysCon), 2013 IEEE International*, pp. 448–451, IEEE, 2013.

[12] J.-O. Lee, T. Kang, K.-H. Lee, S. K. Im, and J. Park, "Vision-based indoor localization for unmanned aerial vehicles," *Journal of Aerospace Engineering*, vol. 24, no. 3, pp. 373–377, 2010.

[13] C. Mostegel, A. Wendel, and H. Bischof, "Active monocular localization: Towards autonomous monocular exploration for multirotor mavs," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 3848–3855, IEEE, 2014.

[14] G. Tuna, K. Gulez, V. C. Gungor, and T. V. Mumcu, "Evaluations of different simultaneous localization and mapping (slam) algorithms," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pp. 2693–2698, IEEE, 2012.

[15] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[16] A. Huletski, D. Kartashov, and K. Krinkin, "Evaluation of the modern visual slam methods," in *Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015*, pp. 19–25, IEEE, 2015.

[17] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pp. 155–160, IEEE, 2011.

[18] A. Breitenmoser, L. Kneip, and R. Siegwart, "A monocular vision-based system for 6d relative robot localization," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 79–85, IEEE, 2011.

[19] S. Lupashin, A. Schöllig, M. Hehn, and R. D'Andrea, "The flying machine arena as of 2010," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 2970–2971, IEEE, 2011.

[20] A. Milella, D. Di Paola, G. Cicirelli, and T. D'Orazio, "Rfid tag bearing estimation for mobile robot localization," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, IEEE, 2009.

[21] Vicon, "Vicon motion capture system," 2017. [Online; accessed September 8, 2017].

[22] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, pp. 72–77, IEEE, 2002.

[23] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for mavs using one single circular landmark," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 495–512, 2011.

[24] A. D. Swart, *Monocular vision assisted autonomous landing of a helicopter on a moving deck*. PhD thesis, Stellenbosch: Stellenbosch University, 2013.

[25] D. F. Malan, *3D tracking between satellites using monocular computer vision*. PhD thesis, Stellenbosch: University of Stellenbosch, 2005.

[26] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[27] Z. Kurt-Yavuz and S. Yavuz, "A comparison of ekf, ukf, fastslam2. 0, and ukf-based fastslam algorithms," in *Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on*, pp. 37–43, IEEE, 2012.

[28] C. Stachniss, "Extended information filter," 2017. [Online; accessed September 19, 2017].

[29] S. Konatowski, P. Kaniewski, and J. Matuszewski, "Comparison of estimation accuracy of ekf, ukf and pf filters," *Annual of Navigation*, vol. 23, no. 1, pp. 69–87, 2016.

[30] A. Mordvintsev and A. K, "Changing colorspaces," 2013. [Online; accessed September 19, 2017].

[31] A. Mordvintsev and A. K, "Understanding features," 2013. [Online; accessed September 20, 2017].

[32] A. Chiu, *Probabilistic Outlier Removal for Stereo Visual Odometry*. PhD thesis, Stellenbosch: Stellenbosch University, 2017.

[33] A. Mordvintsev and A. K, "find chessboard corners," 2013. [Online; accessed September 20, 2017].

[34] P. Abeles, "Camera calibration," 2017. [Online; accessed September 20, 2017].

[35] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[36] opencv dev team, "Pinhole camera model," 2017. [Online; accessed September 20, 2017].

[37] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2969–2976, IEEE, 2011.

[38] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[39] T. Nöll, A. Pagani, and D. Stricker, "Markerless camera pose estimation-an overview," in *OASIcs-OpenAccess Series in Informatics*, vol. 19, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[40] opencv dev team, "Solve pnp," 2017. [Online; accessed September 20, 2017].

[41] H. Pienaar and H. C. Reader, "Multicopter metrology platform for propagation measurements," in *Electromagnetics in Advanced Applications (ICEAA), 2015 International Conference on*, pp. 370–374, IEEE, 2015.

[42] u-Blox, *u-blox M8 concurrent GNSS modules*, Oct. 2015.

[43] InvenSense, *MPU-6000 and MPU-6050 Product Specification*, Aug. 2013.

[44] RaspberryPi.org, "Camera module," 2017. [Online; accessed September 21, 2017].

[45] Amazon, "Raspicam camera board module," 2017. [Online; accessed September 21, 2017].

[46] Swift Navigation, *Piksi Datasheet*, 2016.

[47] I. Peddle and J. Engelbrecht, "Introductory course to aircraft dynamics," *Advanced Automation*, vol. 833.

[48] http://www.starlino.com, "Dcm tutorial," 2011. [Online; accessed August 9, 2017].

[49] http://opencv-python-tutroals.readthedocs.io, "Camera calibration," 2013. [Online; accessed August 14, 2017].

[50] IkamusumeFan, "Multivariate normal sample," 2014. [Online; accessed October 11, 2017].

[51] lidar-uk.com, "The science and technology behind lidar," 2013. [Online; accessed September 6, 2017].